

BEST AVAILABLE COPY

ESTIMATING GRAPHICS SYSTEM PERFORMANCE FOR POLYGONS

Patent number: JP2002503854T

Publication date: 2002-02-05

Inventor:

Applicant:

Classification:

- international: G06T15/00; G09G5/36

- european:

Application number: JP20000531810T 19990216

Priority number(s): US19980074838P 19980217; WO1999US03227
19990216

Also published as:



WO9941704 (A1)

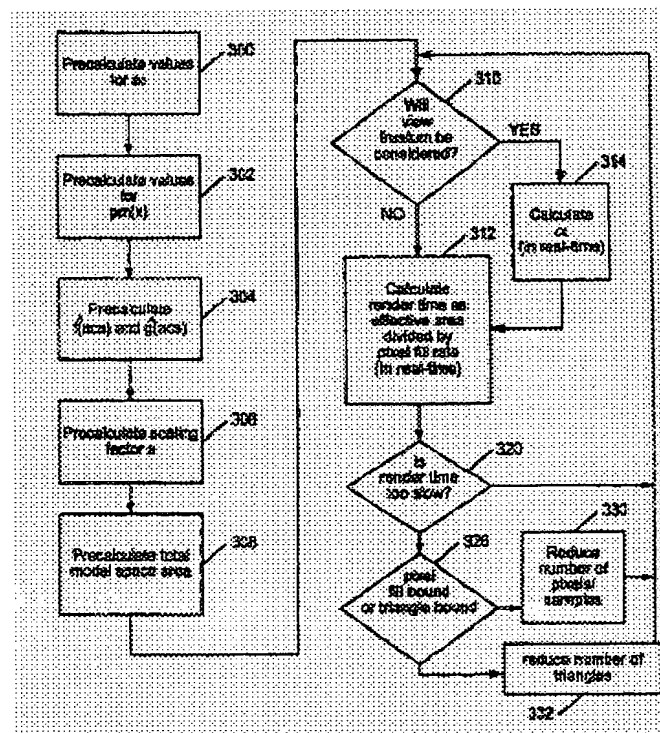
EP1055199 (A1)

EP1055199 (B1)

Abstract not available for JP2002503854T

Abstract of correspondent: **WO9941704**

A method for estimating rendering times for three-dimensional graphics objects and scenes is disclosed. The rendering times may be estimated in real-time, thus allowing a graphics system to alter rendering parameters (such as level of detail and number of samples per pixel) to maintain a predetermined minimum frame rate. Part of the estimation may be performed offline to reduce the time required to perform the final estimation. The method may also detect whether the objects being rendered are pixel fill limited or polygon overhead limited. This information may allow the graphics system to make more intelligent choices as to which rendering parameters should be changed to achieve the desired minimum frame rate. A software program configured to efficiently estimate rendering times is also disclosed.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号
特表2002-503854
(P2002-503854A)

(43) 公表日 平成14年2月5日 (2002.2.5)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 T 15/00	1 0 0	G 0 6 T 15/00	1 0 0 A 5 B 0 8 0
G 0 9 G 5/36		G 0 9 G 5/36	5 2 0 D 5 C 0 8 2

審査請求 未請求 予備審査請求 有 (全 95 頁)

(21) 出願番号 特願2000-531810 (P2000-531810)
(86) (22) 出願日 平成11年2月16日 (1999.2.16)
(85) 翻訳文提出日 平成12年8月17日 (2000.8.17)
(86) 国際出願番号 PCT/US99/03227
(87) 国際公開番号 WO99/41704
(87) 国際公開日 平成11年8月19日 (1999.8.19)
(31) 優先権主張番号 60/074, 838
(32) 優先日 平成10年2月17日 (1998.2.17)
(33) 優先権主張国 米国 (US)

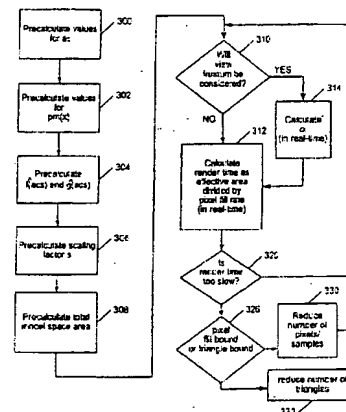
(71) 出願人 サン・マイクロシステムズ・インコーポレ
ーテッド
SUN MICROSYSTEMS, IN
CORPORATED
アメリカ合衆国 94303 カリフォルニア
州・パロ アルト・サン アントニオ ロ
ード・901
(72) 発明者 ディアリング, マイケル・エフ
アメリカ合衆国・94024・カリフォルニア
州・ロス アルトス・クエスタ ドライ
ブ・657
(74) 代理人 弁理士 山川 政樹

最終頁に続く

(54) 【発明の名称】 ポリゴンのためのグラフィックス・システム・パフォーマンスの推定

(57) 【要約】

三次元グラフィックス・オブジェクトおよび画面のレンダリング時間を推定する方法を開示する。レンダリング時間は、リアルタイムで推定することができ、したがって、グラフィックス・システムはレンダリング・パラメータ (詳細度や1ピクセル当たりのサンプル数など) を変更して所定の最低フレーム・レートを維持することができる。最終計算を行うのに要する時間を短縮するために、この推定の一部はオフラインで行われる。この方法は、レンダリングしているオブジェクトがピクセル・フィル制限付きであるかポリゴン・オーバーヘッド制限付きであるかを検出することもできる。この情報によって、グラフィックス・システムは所望の最低フレーム・レートを実現するためにどのレンダリング・パラメータを変更すべきであるかに関して賢い選択を行うことができる。レンダリング時間を効率的に推定するように構成されたソフトウェア・プログラムも開示する。



この図は、ポリゴン・オブジェクトのレンダリング時間を推定する方法を示す。図中の各ステップは、以下の通りである。

- 300: 事前計算された値 (a) の取得
- 302: 事前計算された値 (project) の取得
- 304: 事前計算された値 (area and lines) の取得
- 306: 事前計算された値 (scaling factor) の取得
- 308: 事前計算された値 (total screen space area) の取得
- 310: 判断: Will this feature be considered? (この特徴は考慮されるか?)
- 312: 計算: Calculate render time as effective area divided by pixel fill rate (in real time) (有効領域をピクセル充填率で割ってリアルタイムレンダリング時間を計算)
- 314: 計算: Calculate (in real time) (リアルタイム計算)
- 320: 判断: Is render time too slow? (レンダリング時間が遅すぎるか?)
- 322: 処理: Reduce number of pixels/samples (ピクセル/サンプル数を減らす)
- 324: 判断: Total fit bound or triangle bound (総適合境界または三角形境界)
- 326: 判断: Is total fit bound or triangle bound? (総適合境界または三角形境界か?)
- 328: 処理: Reduce number of triangles (三角形数を減らす)

【特許請求の範囲】

【請求項 1】 コンピュータ・ソフトウェア・プログラムがポリゴン・レンダリングのためのグラフィックス・システムのパフォーマンスを推定するように構成され、搬送媒体に記録された、グラフィックス・システムにおけるレンダリング時間を推定するコンピュータ・ソフトウェア・プログラムであって、

グラフィックス・データのセットのレンダリング時間を計算し、前記レンダリング時間が所定の最低フレーム・レートによって指定されたレンダリング時間を超える場合、

前記グラフィックス・データがポリゴン・オーバーヘッド制限付きであるかピクセル・フィル制限付きであるかを判断し、

前記所定の最低フレーム・レートを達成するようにレンダリング・パラメータを変更するように構成された複数の命令を含む、コンピュータ・ソフトウェア・プログラム。

【請求項 2】 グラフィックス・データの前記セットが複数の汎用オブジェクトを含み、各汎用オブジェクトが複数のオブジェクト変形体を含み、各オブジェクト変形体が複数のポリゴンとレンダリング属性とを含み、前記複数の命令が、前記汎用オブジェクトのうちの 1 つまたは複数の汎用オブジェクトのために異なるオブジェクト変形体を選択することによって前記レンダリング・パラメータを変更するようにさらに構成された請求項 1 に記載のコンピュータ・ソフトウェア・プログラム。

【請求項 3】 前記複数の命令が、各汎用オブジェクトの各変形体のレンダリング時間を計算し、レンダリングする各汎用オブジェクトのために前記所定の最低フレーム・レートに整合した視覚的に最も現実感のあるオブジェクトを選択するようにさらに構成された請求項 2 に記載のコンピュータ・ソフトウェア・プログラム。

【請求項 4】 前記複数の命令が、前記所定の最低フレーム・レートとの整合性と、ヒステリシスと、各オブジェクトの相対的重要度と、各オブジェクトの位置とを含むグループから選択された 1 つまたは複数の考慮事項に基づいて、レンダリングする各汎用オブジェクトのために視覚的に最も現実感のあるオブジェ

クト変形体を選択するように構成された請求項2または3に記載のコンピュータ・ソフトウェア・プログラム。

【請求項5】 汎用オブジェクトのサブセットに他の汎用オブジェクトより高い優先順位が与えられる請求項3に記載のコンピュータ・ソフトウェア・プログラム。

【請求項6】 前記複数の命令が、前記オブジェクト変形体のうちの少なくとも1つのオブジェクト変形体の累積確率密度分布 $f(a)$ を計算するようにさらに構成され、 $f(a)$ が a 以下の面積を有する前記オブジェクト変形体内のランダムに選定されたポリゴンの確率である請求項2、3、または5に記載のコンピュータ・ソフトウェア・プログラム。

【請求項7】 前記複数の命令が、前記オブジェクトの変形体のうちの少なくとも1つのオブジェクト変形体内のすべてのポリゴンの累積面積 $g(a)$ を計算するようにさらに構成され、 $g(a)$ が前記オブジェクト変形体の合計表面積に対する、面積 a を有する前記オブジェクト変形体内のポリゴンによって占められる合計表面積の量の割合である請求項6に記載のコンピュータ・ソフトウェア・プログラム。

【請求項8】 前記複数の命令が、複数の異なる視角にわたって $f(a)$ を平均するようにさらに構成された請求項6に記載のコンピュータ・ソフトウェア・プログラム。

【請求項9】 前記複数の命令が、
前記汎用オブジェクトのうちの1つまたは複数の汎用オブジェクトのための可能な視点を複数の異なる視角セットに分け、

異なる各視角セットのための $f(a)$ の平均値を計算し、

前記レンダリング時間の計算に使用するために現行視点に基づいて $f(a)$ の前記平均値のうちの1つの値を選択するようにさらに構成された請求項6に記載のコンピュータ・ソフトウェア・プログラム。

【請求項10】 前記複数の命令が、複数の異なる視角にわたって $g(a)$ を平均するようにさらに構成された請求項7、8、または9に記載のコンピュータ・ソフトウェア・プログラム。

【請求項11】 前記複数の命令が、各ポリゴンのアスペクト比を計算するようにさらに構成され、各ポリゴンの前記アスペクト比が前記ポリゴンの高さを前記ポリゴンの幅で割った割合である請求項7に記載のコンピュータ・ソフトウェア・プログラム。

【請求項12】 前記複数の命令が、各ポリゴンの歪みを計算するようにさらに構成され、各ポリゴンの前記歪みが前記ポリゴンの角幅を前記ポリゴンの幅で割った値である請求項11に記載のコンピュータ・ソフトウェア・プログラム。

【請求項13】 $f(a)$ 、 $g(a)$ 、アスペクト比、および歪みが、モデル空間においてオフラインで1回と画面空間のためにリアルタイムで1回の、それぞれ2回計算される請求項12に記載のコンピュータ・ソフトウェア・プログラム。

【請求項14】 前記複数の命令が、各オブジェクト変形体の複数の異なる視点についてレンダリング時間を計算し、次に、各オブジェクト変形体の前記複数の異なる視点について計算されたレンダリング時間を平均するようにさらに構成された請求項1、2、12、または13に記載のコンピュータ・ソフトウェア・プログラム。

【請求項15】 前記複数の命令が、各オブジェクト変形体の可能なすべての視点についてレンダリング時間を計算し、次に、計算されたレンダリング時間を平均するように構成された請求項14に記載のコンピュータ・ソフトウェア・プログラム。

【請求項16】 前記複数の命令が、前記オブジェクト変形体の対称な部分の代表的な視点のレンダリング時間を計算してから、計算されたレンダリング時間を平均することによって、前記オブジェクト変形体の可能なすべての視点についてレンダリング時間を計算するように構成された請求項15に記載のコンピュータ・ソフトウェア・プログラム。

【請求項17】 前記複数の命令が、畳込み計算を行うことによって前記オブジェクト変形体の可能なすべての視点についてレンダリング時間を計算し、平均するようにさらに構成された請求項15または16に記載のコンピュータ・ソ

フトウェア・プログラム。

【請求項 18】 前記レンダリング・パラメータが、サンプル密度、1 ピクセル当たりのサンプル数、ピクセル数、照明効果、光源数、詳細レベル、ポリゴン数、アンチエイリアシング、フォギング、テクスチャ・マッピング・パラメータ、プログラム可能シェーダ、シェーディング・パラメータ、およびカラー階調から成るグループから選択される請求項 1、2 または 16 に記載のコンピュータ・ソフトウェア・プログラム。

【請求項 19】 前記搬送媒体がコンピュータ可読媒体または伝送媒体である請求項 1 または 2 に記載のコンピュータ・ソフトウェア・プログラム。

【請求項 20】 幾何形状データの特定のセットのためのグラフィックス・システムのレンダリング・パフォーマンスを推定する方法であって、

前記グラフィックス・システムのピクセル・フィル・レートを判断するステップと、

前記グラフィックス・システムの前記幾何形状データの 1 フレーム当たりのレンダリング時間を計算するステップとを含み、前記幾何形状データが複数のポリゴンを含み、前記 1 フレーム当たりのレンダリング時間が前記幾何形状データの有効面積を前記ピクセル・フィル・レートで割った値であり、前記有効面積が前記幾何形状内のすべての正面向きポリゴンの面積の和に等しく、所定の面積 a_c 未満の面積を有する前記幾何形状内の各正面向きポリゴンの面積が前記所定の面積 a_c に切り上げられ、前記所定の面積 a_c が前記グラフィックス・システムのパフォーマンスを記述する定数である方法。

【請求項 21】 前記 1 フレーム当たりのレンダリング時間が背面向きポリゴンのための調整値を含み、前記調整値が、背面向きポリゴンの数に所定の面積 a_c を乗じた近似値および背面処理時間と最小ポリゴン・レンダリング時間との比である請求項 20 に記載の方法。

【請求項 22】 グラフィックス・データの前記セットが複数の汎用オブジェクトを含み、各前記汎用オブジェクトが複数のオブジェクト変形体を含み、前記方法が、前記 1 フレーム当たりのレンダリング・レートが所定の最低値を下回り、前記グラフィックス・システム・パフォーマンスがポリゴン・オーバーヘッド

ド制限付きである場合、レンダリングのために前記オブジェクト変形体のうちのより少ないポリゴン数を有する1つのオブジェクト変形体を選択するステップをさらに含む請求項20または21に記載の方法。

【請求項23】 前記1フレーム当たりのレンダリング時間の結果として所定の最低値を下回るフレーム・レートになり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きである場合、前記グラフィックス・システムにグラフィックス・データの前記セットのピクセル面積を小さくさせるステップをさらに含む請求項21に記載の方法。

【請求項24】 前記グラフィックス・システムがサンプル密度を変えることができ、前記方法が、前記1フレーム当たりのレンダリング時間の結果として所定の最低値を下回るフレーム・レートになり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きである場合、前記グラフィックス・データの少なくとも一部をレンダリングするために使用される前記グラフィックス・システムのサンプル密度を低下させるステップをさらに含む請求項21に記載の方法。

【請求項25】 前記1フレーム当たりのレンダリング・レートが所定の最低値より下がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きである場合、前記グラフィックス・データの少なくとも一部に対して行われるテクスチャ計算の複雑度を低下させることによって前記グラフィックス・データのピクセル面積を減らすステップをさらに含む請求項21または24に記載の方法。

【請求項26】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがポリゴン・オーバーヘッド制限付きでない場合、レンダリングのためのより多いポリゴン数を有するセット・グラフィックス・データを選択するステップをさらに含む請求項21に記載の方法。

【請求項27】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システムがフィル・レート制限付きでない場合、前記グラフィックス・システムに前記グラフィックス・データのピクセ

ル面積を増やさせるステップをさらに含む請求項21または26に記載の方法。

【請求項28】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きでない場合、ビデオ・サイズ変更によって前記グラフィックス・データのピクセル面積を増やすステップをさらに含む請求項21または26に記載の方法。

【請求項29】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きでない場合、1つまたは複数の背景オブジェクトを削除することによって前記グラフィックス・データのピクセル面積を減らすステップをさらに含む請求項21または26に記載の方法。

【請求項30】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きでない場合、前記グラフィックス・データの少なくとも一部をレンダリングするために使用されるサンプル密度を低下させるステップをさらに含む請求項21または26に記載の方法。

【請求項31】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きでない場合、前記グラフィックス・データの少なくとも一部に対して行われるテクスチャ計算の複雑度を低下させることによって前記グラフィックス・データの画素面積を減らすステップをさらに含む請求項21または26に記載の方法。

【請求項32】 グラフィックス・システムが幾何形状データの特定のセットをレンダリングする近似レンダリング時間を予測する方法であって、

前記グラフィックス・システムのピクセル・フィル・レートを判断するステップと、

レンダリングするグラフィックス・データの特定のセットのための有効面積を計算するステップと、

前記有効面積を前記ピクセル・フィル・レートで割ることによって前記幾何形

状のレンダリング時間をリアルタイムで推定するステップとを含む方法。

【請求項33】 前記有効面積を計算する前記ステップが、

実面積が所定の定数 a_c と等しいかまたは上回る前記幾何形状データ内のポリゴンの表面積に対応する、幾何形状データの前記セットのための実面積を計算するステップと、

疑似面積が最小ポリゴン・オーバーヘッドの効果に近似し、前記所定の定数 a_c に前記所定の定数 a_c より小さい前記幾何形状データ内のポリゴンの数を乗じた値に対応する、幾何形状データの前記セットのための疑似面積を計算するステップと、

前記実面積と前記疑似面積とを合計するステップとを含む請求項32に記載の方法。

【請求項34】 前記実面積を計算する前記ステップが、

a_c が所定の定数であり、 s が g^* の計算のためのスケーリング係数とみなされるモデル空間－画面空間スケーリング係数である、 a_c と s との積より小さいかまたは等しい面積を有するポリゴンによって占められる合計表面積を近似する関数 $g^*(a_c s)$ の値を求めるステップと、

幾何形状データの前記セットのための合計画面面積を事前計算するステップと、

モデル空間－画面空間スケーリング係数を計算するステップと、

$(1 - g^*(a_c s))$ 実行数値積分を使用して前記合計面積乗算するステップとを含む請求項33に記載の方法。

【請求項35】 前記推定ステップと前記計算ステップがリアルタイムで行われ、前記判断ステップがオフラインで行われる請求項32、33、または34に記載の方法。

【発明の詳細な説明】

【0001】

(発明の背景)

1. 発明の分野

本発明は、一般には、コンピュータ・グラフィックスに関し、より詳細には、三次元グラフィックス・システムのポリゴン・レンダリング・パフォーマンスの評価に関する。

【0002】

2. 関連技術の説明

コンピュータ・システムは一般に、コンピュータ画面または表示装置上で視覚出力を生成するのにそのグラフィックス・システムに依存している。初期のグラフィックス・システムは、プロセッサが出力として生成したものを受け取り、それを画面上に表示する役割のみを果たしていた。本質的に、それらは単純な変換機構またはインタフェースとして機能した。しかし、最新のグラフィックス・システムは、多大な処理能力を備えたグラフィックス・プロセッサを組み込んでいる。これらは現在では、単純な変換機構というよりも、コプロセッサのように機能している。この変化は、表示装置に送られるデータの複雑さと量の両方が、最近増大したことによる。たとえば、最新のコンピュータ表示装置は、初期のモデルよりもより多くのピクセルと、より大きな色の深さと、より高いリフレッシュ・レートとを備える。同様に、表示される画像は、現在では、より複雑であり、アンチエイリアシングやテクスチャ・マッピングなど、高度なレンダリング技法を必要とすることがある。

【0003】

その結果、グラフィックス・システムにおける多大な処理能力がなかったとしたら、CPUはグラフィックス計算を行うのに多くの時間を費やすことになる。これによって、プログラムの実行に付随する他のタスクを行うのに必要な処理能力をコンピュータ・システムから奪い、それによって全体的なシステム・パフォーマンスを大幅に低下させる可能性がある。しかし、高性能グラフィックス・システムにより、CPUが画面上にボックスを描画する必要がある場合に、CPU

は各ピクセルの位置と色を計算する必要がなくなっている。その代わり、CPUは、「これらの座標にボックスを描画せよ」という要求をビデオ・カードに送る。すると、グラフィックス・システムがそのボックスを描画し、その結果、プロセッサを他のタスクを行えるように解放する。

【0004】

一般に、コンピュータにおけるグラフィックス・システム（グラフィックス・アクセラレータとも呼ぶ）は、パフォーマンス・レベルを向上させるそれ自体のプロセッサを含むビデオ・アダプタの1タイプである。これらのプロセッサは、グラフィカル変換の計算を行うように特化されており、したがって、コンピュータ・システムによって使用される汎用CPUよりも良い結果を達成する傾向がある。さらに、これらは、グラフィックス・システムがグラフィックス計算を処理している間に、コンピュータのCPUを他のコマンドを実行するように解放する。グラフィカル・アプリケーションと、特にマルチメディア・アプリケーションの普及により、高パフォーマンスなグラフィックス・システムがコンピュータ・システムの一般的機能になっている。ほとんどのコンピュータ製造業者は、現在では、自社のシステムに高パフォーマンス・グラフィックス・システムをバンドルしている。

【0005】

グラフィックス・システムは一般に、限られた機能のみを実行するため、カスタマイズすることができ、したがって、グラフィックス操作ではコンピュータの汎用中央処理装置よりもはるかに効率的である。初期のグラフィックス・システムは二次元（2D）グラフィックスの実行に限られていたが、現在ではその機能は進歩して、シェーディング、フォギング、アルファ・ブレンディング、鏡面ハイライトなどの三次元（3D）グラフィックス操作を含めて、3Dグラフィックス・レンダリング機能も備えるようになっている。

【0006】

3Dグラフィックス・システムの処理能力は、猛烈な速さで向上している。2、3年前は、単純なオブジェクトの陰影付き画像を每秒2、3フレームの割合でレンダリングすることができるに過ぎなかったが、現在のシステムは、複雑なオ

プロジェクトの60Hz以上のレンダリングをサポートする。このより高いパフォーマンスによって、最新のグラフィックス・アプリケーションは、表示されているシーンの現実感を向上させることができる。

【0007】

三次元レンダリング・オブジェクトの現実感を増す1つの一般的な方法は、そのオブジェクトを表示するのに使用されるポリゴンの数を増やすことである。ほとんどの最新のグラフィックス・システムは、まずオブジェクトの表面を複数のポリゴン（すなわち直線によって境界を区切られた密接した平面図形）に分割または「モザイク化（tessellating）」することによって、オブジェクトをレンダリングする。次に、各ポリゴンを個別にレンダリングする。レンダリングは一般に、以下のステップを必要とする。すなわち、（1）ポリゴンの各頂点のためにいくつかのパラメータを計算し、（2）それらの頂点パラメータから補間してポリゴン内を埋める。頂点パラメータの例としては、色情報、透明度情報、奥行情報、ライティング情報、テクスチャ情報がある。

【0008】

オブジェクトを表示するのに使用されるポリゴンの数を増やすことによって、オブジェクトはより滑らかに見え、より現実感のあるテクスチャ外観を持つことができる。図1Aに、第1の数のポリゴンにモザイク化された球形を示す。図1Bに、より多数のポリゴンにモザイク化された同じ球形の一例を示す。これらの図に示すように、より多数のより小さなポリゴンを使用することによって、より現実感のあるシーンをレンダリングすることができる。すべてのポリゴンは一般にレンダリングのために三角形に分割されるため、本明細書では「ポリゴン」という用語と「三角形」という用語を交換可能に使用することに留意されたい。

【0009】

この技法によって現実感は増すが、グラフィックス・システム上の処理負荷も増大する。以前のグラフィックス・アプリケーションは、多数のピクセルを含む大きなポリゴンを使用していた。したがって、各ポリゴンをセットアップする「オーバーヘッド」はグラフィックス・システムの全体的な処理資源のうちの比較的わずかな部分を費やすのに対し、ポリゴン内のピクセルを補間するプロセスは

グラフィックス・システムの処理能力の大部分を必要とする。これらのシステムは、パフォーマンス制限要因が、グラフィックス・システムが計算可能なピクセル数であるため、「ピクセル・フィル制限付き」と呼ぶ。しかし、最新のアプリケーションは、現在では1または2ピクセル（または場合によっては1ピクセルより少ないことさえある）しか含まないポリゴンを使用している。したがって、ポリゴンをセットアップする作業には、実際のピクセル計算プロセスよりも多くの時間を要することがある。これらのシステムは、ポリゴンのセットアップに付随するオーバーヘッドがパフォーマンス制限要因であるため、「ポリゴン・オーバーヘッド制限付き」と呼ぶ。特定のグラフィックス・システムは特定のシーン（たとえば多くの小さなポリゴンのあるシーン）に対してポリゴン・オーバーヘッド制限付きであり、異なるシーン（たとえばより大きなポリゴンまたはより複雑なピクセル・レベルのエンハンスメントのあるシーン）に対してはピクセル・フィル制限付きである場合があることに留意されたい。

【0010】

図2は、グラフィックス・システムの考えられる1つのパフォーマンス限界曲線を示すグラフである。図に示すように、ポリゴン面積が特定のサイズ a_c より下がると、システムのパフォーマンスはポリゴン・オーバーヘッドによって制限される。同様に、ポリゴン・サイズが a_c を超えると、パフォーマンスは最大ピクセル・フィル・レートによって制限される。

【0011】

前述のように、グラフィックス・システムの処理能力は、最近2、3年間に急激に向上した。しかし、このように大幅に向上しても、新しいアプリケーションはさらに高いパフォーマンスを必要とし続ける。たとえば、ある種のコンピュータ・ゲームや仮想現実プログラムは、複数の複雑な三次元オブジェクトのリアルタイム・レンダリングを高フレーム・レートで必要とする。これらのグラフィックス多用アプリケーションは、グラフィックス・システムのパフォーマンスに対する要求が高く、グラフィックス・システムの能力をたやすく超える。

【0012】

1つの可能性は、アプリケーションがグラフィックス・システムのパフォーマ

ンス能力を超える場合、フレーム・レートを下げることである。しかし、グラフィックス・アプリケーションによっては、それを下回るとそのアプリケーションが使用不能になる最低フレーム・レートを有するものもあるため、これは常に可能であるとは限らない。たとえば、3Dコンピュータ・ゲームのフレーム・レートが特定のレベルを下回る場合、画面上の動きや動画がぎくしゃくするようになる。さらに、フレーム・レートが特定のレベルより下がった場合、ユーザが1つの動作（たとえばミサイルの発射など）を行ったときと、その動作のグラフィックス表現が画面上に現れるときとの間の遅延が長くなり過ぎて、ゲームが遊べなくなる。したがって、多くのアプリケーションでは、レンダリングするシーンの複雑さが大幅に増大する場合であっても、フレーム・レートを特定の所定レベルより下げることができない。

【0013】

しかし、それに対して、システムがポリゴン・オーバーヘッド制限付きである場合、フレーム・レートの低減が持つ制限や欠点なしに、表示されるポリゴンの数を調整することができる。たとえば、（ポリゴン・オーバーヘッドのために）特定のフレーム・レートで1フレームにつき最大1000個のポリゴンをレンダリングすることができる特定のグラフィックス・システムを仮定し、単一の移動する球体を表示する場合、その球体はそのフレーム・レートに影響を与えずに1000個のポリゴンにしか分割する（すなわち「モザイク化する」）ことができない。2つの類似した移動する球体を表示する場合、グラフィックス・システムは、（前述のマイナスの副次効果を伴って）フレーム・レートを半分にするか、各球体を500個のみのポリゴンを使用して表示することができる。後者の選択肢を使用すると、球体はよりぎざぎざに見えるようになる（すなわち球体の表面がなめらかに見えなくなる）が、その動きにはぎこちなさは加えられない。この技法を、詳細レベル（LOD）の低減とも呼ぶことができる。

【0014】

システムがピクセル・フィル制限付きである場合、フレーム・レートの低減に代わる他の代替策は、レンダリングする全ピクセル数を減らすことである。これは、所与のオブジェクトのピクセル面積を減らすか、表示されるオブジェクトの

数を減らすか、またはシーン全体内のピクセル数を減らすことによって行うことができる。

【0015】

グラフィックス・システムはポリゴンの最適な大きさおよび数と、ピクセルの最適数とを正確に判断することができれば理想的である。これによって、グラフィックス・システムは、フレーム・レートを下下させず、ポリゴン・オーバーヘッドでパフォーマンスを無駄に使用せずに、最大数のポリゴンを含めるように表示シーンを最適化することができるであろう。

【0016】

したがって、特定のシーンのパフォーマンス限界を効率的に判断することができるグラフィックス・システムが望まれる。同様に、グラフィックス・システムのパフォーマンス限界を効率的に判断する方法も望まれる。

【0017】

(発明の概要)

上記で概説した問題は、幾何形状データの特定のセットのレンダリング・パフォーマンスを推定するように構成されたグラフィックス・システムによってある程度解決することができる。ある種の実施態様では、グラフィックス・システムは、シーン・レンダリング時間をフレームごとに推定し、次に、レンダリング・パラメータ（たとえばポリゴン数、ピクセル数、サンプル数、またはフィーチャ数）を調整して望ましい最低フレーム・レートを維持する。

【0018】

一実施態様では、グラフィックス・システムは、幾何形状データ内の各ポリゴンごとに「有効ポリゴン面積」を計算することによってレンダリング・パフォーマンスを推定することができる。有効ポリゴン面積は、システムのピクセル・フィル制限とポリゴン・オーバーヘッド制限の観点からポリゴンがグラフィックス・システムに与える影響を考慮に入れたポリゴンのレンダリング時間の推定値である。グラフィックス・システムは、全ポリゴンの有効面積の総和を推定して、総「有効面積」を推定することができる。この有効面積は、モデル空間と画面空間の両方について計算することができる。本明細書で使用する「モデル空間」と

は、幾何形状データが指定されている座標系を指し、「画面空間」とは、表示装置上のピクセルによって規定された座標系を指す。同様に、「幾何形状データ・セット」とは、グラフィックス・システムが受け取って1つまたは複数のフレームにレンダリングするグラフィックス・データを指す。グラフィックス・データは、データをレンダリング（すなわち描画）するのに十分な情報をグラフィックス・システムに与える頂点および／または命令（たとえば命令コード）を含むことができる。グラフィックス・データは、レンダリングする2Dと3Dの両方のオブジェクトの組合せを表すことができる。

【0019】

ある種の実施態様では、有効面積は、ポリゴンの裏側を向いた面を全部または一部含むことができる。これらの実施態様は、相当なシステム処理資源を使用して背面向きポリゴンをカリング（culling）するグラフィックス・システムについて、より正確な推定値を生成することができる。

【0020】

さらに、有効面積は、所定の大きさより小さいポリゴンの「疑似面積」値を含むことができる。疑似面積とは、ポリゴンの一部としてレンダリングした場合、所定の大きさより小さいポリゴンのオーバーヘッド処理時間に近くなる追加の理論上の面積を指す。小さいポリゴンのオーバーヘッド処理時間を補償する他の方法は、所定の臨界面積（本明細書では「 a_c 」と呼ぶ）を下回るすべてのポリゴンを単に臨界面積に切り上げることである。値 a_c は、それを下回るとポリゴン・セットアップ・オーバーヘッドが有意な制限要因になる、ポリゴンの最小サイズを表す。たとえば、大きなポリゴン（たとえば面積40ピクセル）は、セットアップに2クロック・サイクルしか要せず、レンダリングに40クロック・サイクル要する。この場合、ポリゴンをレンダリングするのに要する時間と比較して、ポリゴンのセットアップに付随するオーバーヘッドは比較的少ない。しかし、小さなポリゴン（たとえば1ピクセル面積またはサブピクセル面積）はセットアップにやはり2クロック・サイクルを要するが、レンダリングには1クロック・サイクルしか要しない。したがって、より小さいポリゴンの場合、ポリゴンをセットアップするオーバーヘッドは、重大なパフォーマンス制限になる可能性がある。

る。より小さいポリゴンを所定の臨界面積 a_c に切り上げることによって、小さいポリゴンのセットアップ・オーバーヘッドの影響を計算することができる。同様に、小さいポリゴンの面積に疑似面積を加えることも同じ機能を果たす。値 a_c はグラフィックス・システムによって異なり、グラフィックス・システムの特定の構成（たとえばカラー階調）に応じて変わる場合があることに留意されたい。

【0021】

a_c の事前計算値を使用して特定の幾何形状データ・セットがポリゴン・オーバーヘッド限界とピクセル・フィル限界のいずれになるかを判断するようにシステムを構成することができる。前述のように、ポリゴン・オーバーヘッド限界という用語は、グラフィックス・システムのパフォーマンスが1ポリゴン当たりの処理（たとえばライティング）によって制限される場合を指す。それに対して、ピクセル・フィル限界という用語は、グラフィックス・システムのパフォーマンスが1ピクセル当たりの計算（たとえば透明性、テクスチャリング、およびアンチエイリアシング）または帯域幅によって制限される場合を指す。

【0022】

一実施態様では、グラフィックス・システムは、これらの計算をリアルタイムまたはニア・リアルタイムで行うことができる。本明細書では、タスクが平均的ユーザにわかる（たとえば1フレームごとまたは1表示装置リフレッシュ・サイクルごとの）遅延を生じさせずに行われる場合、タスクは「リアルタイム」で実行されるという。逆に、本明細書では、タスクがリアルタイムで実行されない場合（すなわちユーザにわかる遅延を生じさせる場合）、タスクは「オフライン」で実行されるという。

【0023】

ある種の実施態様では、レンダリングした場合に特定のグラフィックス・データ・セットがポリゴン・オーバーヘッド限界またはピクセル・フィル限界になると判断し、グラフィックス・システムのフレーム・レートが所定のしきい値より下がると判断した後、グラフィックス・システムは、フレーム・レートをしきい値より上に上げるために、レンダリングしているシーンまたはレンダリング・パ

ラメータに動的に変更を加える。たとえば、グラフィックス・データのセットがピクセル・フィル限界である場合、グラフィックス・システムはそのシーン内のサンプルまたはピクセルの数または密度を減らすように構成することができる。あるいは、グラフィックス・システムは、レンダリングしているオブジェクトまたは画像の全体的サイズを小さくすることができる。

【0024】

それに対して、グラフィックス・データのセットがポリゴン・オーバヘッド限界であり、システムのフレーム・レートが所定のしきい値より下がる場合、グラフィックス・システムは、より大きいポリゴンを使用してシーンまたはオブジェクトをモザイク化することによって詳細レベル（LOD）を下げるか、またはグラフィックス・システムは、より少ないポリゴンを持つグラフィックス・データの事前モザイク化されたセットを選択することができる。表示されるオブジェクトの数を少なくすることによって、フレーム・レートをさらに向上させることができる。

【0025】

ある種の実施態様では、グラフィックス・システムは「累積確率分布」を計算して、レンダリングするシーンまたはオブジェクトのパフォーマンス限界のリアルタイム計算をサポートする。累積確率分布は、特定の幾何形状データ・セット内のポリゴンサイズの累積分布を表す。累積確率値分布は、（1） a_c 以下の面積を有する無作為に選択したポリゴンの確率（ $f(a_c)$ と呼ぶ）、または（2）面上の無作為に選択した点が a_c 以下の面積を有するポリゴンに属する確率（ $g(a_c)$ と呼ぶ）としての2つの形態で計算することができる。関数 $f(a_c)$ および $g(a_c)$ は、画面空間について、モデル空間ポリゴン画面空間ポリゴンに変換するときに行われるスケーリングを示すスケーリング係数を表す所定の定数を適用することによって計算することもできる。これらの画面空間累積確率分布関数を、本明細書ではそれぞれ $f^{\sim}(a_c s)$ および $g^{\sim}(a_c s)$ と呼ぶ（本訳文においては「 \sim 」は実際にはその前のローマ字の上に付く。）。

【0026】

幾何形状レンダリング時間のリアルタイムまたはニア・リアルタイム推定を可

能にするため、オフライン方式で前もっていくつかの予備計算を行うことができる。これらの計算は、リアルタイムまたはニア・リアル・タイム方式で生成することもできるが、これらをオフラインで行うことによって計算の残りのリアルタイム部分の待ち時間をさらに短縮することができる。たとえば、(グラフィックス・システムの異なる構成に対応する) a_c のいくつかの異なる値をオフラインで計算することができる。関数 $p_m(x)$ もオフラインで計算することができる。ただし、 $p_m(x)$ はディラック・デルタ関数 $p_m(a) = \delta(A - a)$ によるモデル空間確率分布であり、 A はモデル空間内の単一のポリゴンの面積である。

【0027】

システムは、以下の数式に従った数値積分によって、 $\hat{f}(a_c s)$ および $\hat{g}(a_c s)$ もオフラインで計算することができる。

【数1】

$$\hat{f}(a) = \int_0^a p_m(x) dx + \int_a^\infty \frac{a}{x} \cdot p_m(x) dx,$$

$$\text{ただし } f(a) = \hat{f}(a_c s)$$

、ならびに、

【数2】

$$\hat{g}(a) = \int_0^\infty \int_0^{\min(y, a)} 2 \cdot \frac{x^2}{y^2} \cdot p_m(y) dx dy,$$

$$\text{ただし } g(a) = \hat{g}(a_c s)$$

【0028】

a_c 、 $\hat{f}(a_c s)$ 、および $\hat{g}(a_c s)$ をオフラインで計算することにより、(1) モデリング行列およびビューイング行列からスケーリング係数 s を計算し、(2) 以下の数式に従ってレンダリング・レートを求めることによって、グラフィックス・データをレンダリングするのに要する時間の推定を効率的にリアルタイムで行うことができる。

【数3】

$$\begin{aligned} \text{レンダリング時間} &= \frac{\text{有効面積}}{\text{ピクセル・フィル・レート}} \\ &= \frac{(a_0 \cdot n \cdot \hat{f}(a_0, s) + 1) + \left[\frac{1}{2} \cdot s \cdot \text{合計モデル空間面積} \right] \cdot (1 - \hat{g}(a_0, s))}{\text{ピクセル・フィル・レート}} \end{aligned}$$

上式で、項

【数4】

$$\left[\frac{1}{2} \cdot s \cdot \text{合計モデル面積} \right]$$

は合計画面面積を近似するために使用される。

待ち時間をさらに短縮するために、合計モデル空間面積も事前計算することができることに留意されたい。

【0029】

一実施態様では、フラスタム (frustum: 切頭体) ・クリッピングの可能性に対処することによって、レンダリング時間推定値をさらに向上させることができる。フラスタム・クリッピングとは、表示装置の可視域内にないポリゴン (たとえば画面から外れたポリゴン) をカリング除去する処理を指す。現在のビュー・フラスタムの外部にあるポリゴンの部分の推定値を表すために値 a をリアルタイムで計算することができる。計算した後は、この値 a を以下のようにレンダリング時間推定値に組み込むことができる。

【数5】

$$\text{レンダリング時間} = \frac{(1 - \alpha) \cdot \text{有効面積} + \alpha \cdot a_0}{\text{ピクセル・フィル・レート}}$$

【0030】

a はいくつかの異なる方法で計算することができるが、1つの単純な推定は、オブジェクトの境界ボックスを調べ、次に境界ボックスのどの部分が表示可能領域の外部に入るかを判断することによって得られる。より正確な推定値の場合、

複数のより小さな境界ボックスをオブジェクトのポリゴンに使用することができる。

【0031】

本発明の上記およびその他の目的、特徴、および利点は、添付図面と共に以下の詳細な説明を読めばより完全に理解することができる。

【0032】

本発明は、様々な変更および代替態様が可能であるが、本発明の特定の実施形態を例として図面に示し、本明細書で詳述する。しかし、図面および詳細な説明は、本発明を開示の特定の態様に限定することを意図したものではなく、その意図は、特許請求の範囲によって規定されている本発明の主旨および範囲内に入るすべての変更、同等物、および代替物を含めることであることを理解されたい。

【0033】

(いくつかの実施形態の詳細な説明)

コンピュータ・システム—図3

次に図3を参照すると、三次元(3-D)グラフィックス・システムを含むコンピュータ・システム80の一実施形態が図示されている。この3-Dグラフィックス・システムは、コンピュータ・システム、ネットワークPC、インターネット機器、HDTVシステムおよびインタラクティブ・テレビジョン・システムを含むテレビジョン、パーソナル・デジタル・アシスト(PDA)、フライト・シミュレータ、自動車運転シミュレータ、船舶シミュレータ、仮想現実環境、および2Dおよび3Dグラフィックスを表示するその他の装置などを含めて、様々なシステムのいずれにも組み込むことができる。

【0034】

図のように、コンピュータ・システム80は、システム・ユニット82と、そのシステム・ユニット82に結合されたビデオ・モニタまたは表示装置84とを含む。表示装置84は、様々なタイプの表示モニタまたは表示装置(たとえばCRT、LCD、またはガスプラズマ表示装置)のいずれでもよい。コンピュータ・システムには、キーボード86および/またはマウス88、またはその他の入力装置(たとえばトラックボール、ディジタイザ、またはタブレット)を含めて

、様々な入力装置を接続することができる。コンピュータ・システム80によってアプリケーション・ソフトウェアを実行して、表示装置84上に3-Dグラフィカル・オブジェクトを表示することができる。以下で詳述するように、一実施形態では、コンピュータ・システム80における3-Dグラフィックス・システムは、ポリゴン・レンダリング・パフォーマンスを効率的に推定し、レンダリング・パラメータを動的に調整して、表示装置84上に表示される画像のフレーム・レート、品質、および現実感を向上させるように構成することができる。

【0035】

コンピュータ・システム・ブロック図ー図4

次に図4を参照すると、図3のコンピュータ・システムを示す略ブロック図が示されている。本発明を理解するのに不要なコンピュータ・システムの要素は、便宜上、図示されていない。図のように、コンピュータ・システム80は、ホスト・バス104とも呼ぶ高速メモリ・バスまたはシステム・バス104に結合された中央処理装置(CPU)102を含む。高速バス104にはシステム・メモリ106も結合することができる。

【0036】

ホスト・プロセッサ102は、たとえばマイクロプロセッサやマルチプロセッサ、CPUなど、様々なタイプの1つまたは複数のプロセッサを含むことができる。システム・メモリ106は、ランダム・アクセス・メモリ(たとえばスタティック・ランダム・アクセス・メモリまたは「SRAM」、シンクロナス・ダイナミック・ランダム・アクセス・メモリまたは「SDRAM」、およびラムバス・ダイナミック・アクセス・メモリまたは「RDRAM」など)および大容量記憶装置を含めて、様々なタイプのメモリ・サブシステムの任意の組み合わせを含むことができる。システム・バスまたはホスト・バス104は、(ホスト・プロセッサとCPUとメモリ・サブシステムとの間の通信のための)1つまたは複数の通信バスまたはホスト・コンピュータ・バスと専用サブシステム・バスとを含むことができる。

【0037】

本発明による3-Dグラフィックス・システムまたはグラフィックス・システ

ム112は、高速メモリ・バス104に結合される。3-Dグラフィックス・システム112は、たとえば、クロス・バー・スイッチまたはその他のバス接続ロジックによって、バス104に結合することができる。高速メモリ・バス104には、他の様々な周辺装置、またはその他のバスも接続することができるものとする。3-Dグラフィックス・システムは、コンピュータ・システム80内のバスのうちの1つまたは複数のバスおよび／または様々なタイプのバスに結合可能であることに留意されたい。さらに、3-Dグラフィックス・システムは、通信ポートにも結合することができ、それによって、外部供給源、たとえばインターネットまたはネットワークからグラフィックス・データを直接受け取ることができる。図に示すように、3-Dグラフィックス・システム112には表示装置84が接続されている。

【0038】

ホストCPU102は、プログラム式入力／出力（I/O）プロトコルに従ってホスト・バス104を介してグラフィックス・システム112との間で情報を送受信する。あるいは、グラフィックス・システム112は、直接メモリ・アクセス（DMA）プロトコルに従って、またはインテリジェント・バス・マスタリングを介して、メモリ・サブシステム106にアクセスすることができる。一実施形態では、ホストCPU102は、上述の計算を行って、（1）レンダリング中のシーンによってフレーム・レートが所定の最低しきい値より下がるか否かを判断し、次に（2）そのシーンがピクセル・フィル制限付きとポリゴン・オーバーヘッド制限付きのいずれであるかに応じて、レンダリング・パラメータを変えるように構成することができる。

【0039】

OpenGLなどのアプリケーション・プログラミング・インタフェース（API）に準拠したグラフィックス・アプリケーション・プログラムをホストCPU102上で実行して、表示装置84上に出力するポリゴンなどの幾何形状プリミティブ（グラフィックス・データ）を定義するコマンドおよびデータを生成することができる。使用している特定のグラフィックス・インタフェースによる定義に従って、これらのプリミティブは前面と背面とで別々の色特性を有すること

ができる。ホスト・プロセッサ102は、これらのグラフィックス・データをメモリ・サブシステム106に転送することができる。その後、ホスト・プロセッサ102は、グラフィックス・データをホスト・バス104を介してグラフィックス・システム112に転送するように動作することができる。他の実施形態では、グラフィックス・システム112は、DMAアクセス・サイクルを使用してホスト・バス104を介して幾何形状データ配列を読み込むことができる。他の実施形態では、グラフィックス・システム112は、インテル・コーポレーションが発表したアドバンスド・グラフィックス・ポート（AGP）などの直接ポートを介してシステム・メモリ106に結合することができる。

【0040】

グラフィックス・システムは、ホストCPU102および／またはシステム・メモリ106、その他のメモリ、またはネットワーク、たとえばインターネットなどの外部供給源から、または放送媒体、たとえばテレビジョンから、またはその他の供給源を含めて、様々な供給源のいずれかからグラフィックス・データを受け取ることができる。

【0041】

後述するように、グラフィックス・システム112は、より効率的なマイクロコード制御を可能にするように構成することができ、その結果、ホスト・プロセッサ102によって生成されたポリゴンに対応する着信カラー値を処理するパフォーマンスが向上する。グラフィックス・システム112はコンピュータ・システム80の一部として図示されているが、グラフィックス・システム112は（たとえばそれ自体の組み込み表示装置を備えた）スタンドアロン装置として構成することもできることに留意されたい。グラフィックス・システム112は、単一チップ装置として、またはチップ上システムまたはマルチチップ・モジュールとしても構成することができる。

【0042】

グラフィックス・システム—図5

次に図5を参照すると、グラフィックス・システム112の一実施形態の詳細を示すブロック図が示されている。図に示すように、グラフィックス・システム

112は、1つまたは複数のグラフィックス・プロセッサ90と、1つまたは複数のスーパーサンプル型サンプル・バッファ162と、1つまたは複数のサンプル・ピクセル計算ユニット170A~Dとを含むことができる。グラフィックス・システム112は、1つまたは複数のデジタル・アナログ変換器(DAC)178A~Bも含むことができる。一実施形態では、グラフィックス・プロセッサ90は、1つまたは複数のレンダリング・ユニット150A~Dを含むことができる。しかし、図の実施形態では、グラフィックス・プロセッサ90は、1つまたは複数の制御ユニット140と、一つまたは複数のデータ・メモリ152A~Dと、1つまたは複数のスケジュール・ユニット154も含む。サンプル・バッファ162は、図に示すように、1つまたは複数のサンプル・メモリ160A~160Nを含むことができる。

【0043】

A. 制御ユニット

制御ユニット140は、グラフィックス・システム112とコンピュータ・システム80との間のデータの伝送を制御することによって、グラフィックス・システム112とコンピュータ・システム80との間のインタフェースとして動作する。2つ以上のレンダリング・ユニット150A~Dを含むグラフィックス・システム112の実施形態では、制御ユニット140はコンピュータ・システム80から受け取ったデータのストリームを、個々のレンダリング・ユニット150A~Dにルーティングされる対応する数の並列ストリームに分割することもできる。グラフィックス・データは、コンピュータ・システム80から圧縮された形で受け取ることができる。これは、コンピュータ・システム80とグラフィックス・システム112との間の帯域幅必要量を削減することができるので有利である。一実施形態では、制御ユニット140は、データ・ストリームを分割し、圧縮された形でレンダリング・ユニット150A~Dにルーティングするように構成することができる。一実施形態では、制御ユニット140は、前述の計算を行って、レンダリングするシーンによってフレーム・レートが所定の最低しきい値より下に下がるか否かを判断し、次にそのシーンがピクセル・フィル制限付きであるかポリゴン・オーバーヘッド制限付きであるかに応じてレンダリング・パ

ラメータを変えるように構成することができる。

【0044】

B. レンダリング・ユニット

レンダリング・ユニット150A～D（本明細書では描画ユニットとも呼ぶ）は、制御ユニット140からグラフィックス命令およびデータを受け取り、次に、その実装態様に応じて、いくつかの機能を実行するように構成される。たとえば、レンダリング・ユニット150A～Dは、グラフィックス・データ内にある様々なグラフィックス・プリミティブの圧縮解除（データが圧縮されている場合）、変換、クリッピング、ライティング、セットアップ、および画面空間レンダリングを実行するように構成することができる。これらの各機能について以下に別々に説明する。

【0045】

受け取った圧縮グラフィックス・データのタイプに応じて、レンダリング・ユニット150A～Dは、算術デコード、ランレングス・デコード、ハフマン・デコード、および辞書デコード（たとえばLZ77、LZSS、LZ78、およびLZW）を実行するように構成することができる。他の実施形態では、レンダリング・ユニット150A～Dは、幾何学圧縮を使用して圧縮されたグラフィックス・データをデコードするように構成することができる。3Dグラフィックス・データの幾何学圧縮は、データ・サイズの大幅な縮小を実現することができると同時に、画像品質のほとんどを保持することができる。3D幾何形状の圧縮と圧縮解除の2つの方法は、米国特許第5793371号、出願第08/511294号（1995年8月4日出願、名称「Method And Apparatus For Geometric Compression Of Three-Dimensional Graphics Data」、出願人整理番号5181-05900）および1998年6月11日出願の「Compression of Three-Dimensional Geometry Data Representing a Regularly Tiled Surface Portion of Graphical Object」という名称の 米国特許出願第09/095777号（出願人整理番号5181-0

6602)に記載されている。圧縮解除をサポートするグラフィックス・システム112の実施形態では、各レンダリング・ユニット150が受け取ったグラフィックス・データは圧縮解除されて1つまたは複数のグラフィックス「プリミティブ」にされ、次にそれをレンダリングすることができる。プリミティブという用語は、オブジェクトの形状（たとえば、二次元または三次元の点、線、三角形、ポリゴン、および三次元の多面体または自由形状面）を定義するオブジェクトの構成要素を指す。

【0046】

変換とは、オブジェクトの操作を指し、オブジェクトの移動（すなわちオブジェクトを異なる場所に移動する）、オブジェクトのスケーリング（すなわち拡大または縮小）、オブジェクトの回転（たとえば三次元空間すなわち「3-空間」において）を含む。

【0047】

クリッピングとは、表示画像の境界線を決め（すなわち、通常は矩形の、クリッピング領域を設定し）、次に、それらの境界線の外部に入るピクセルのレンダリングまたは表示をしないことを指す。

【0048】

ライティングとは、表示画像内のオブジェクトの照明を計算し、個別の各オブジェクトにどのような色および／または輝度を持たせるかを定めることを指す。使用しているシェーディング・アルゴリズム（たとえばコンスタント、グーロー、フォン）に応じて、複数の異なる場所においてライティングを計算しなければならない。たとえば、コンスタント・シェーディングを使用した場合（すなわち、ポリゴンの各ピクセルが同じライティングを有する場合）、ライティングは1ポリゴンにつき1回計算するだけでよい。グーロー・シェーディングを使用した場合、1頂点について1回ライティングを計算する。フォン・シェーディングでは、ピクセルごとにライティングを計算する。

【0049】

セットアップとは、プリミティブを三次元視点にマッピングすることを指す。これは、各オブジェクトをその元のモデル空間座標系からすべてのモデルのため

の「ワールド座標」系への移動と変換を行い、次に確定された視点の座標への移動と変換を行うことを必要とする。これによって、画面上に表示される三次元オブジェクトの正しい透視図が作成される。

【0050】

画面空間レンダリングとは、表示される各ピクセルを生成するために使用されるデータを実際に計算するために行われる計算を指す。従来技術のシステムでは、各ピクセルを計算し、次にフレーム・バッファに格納する。次にフレーム・バッファの内容を表示装置に出力し、最終画像を作成する。しかし、図に示すグラフィックス・システム112の実施形態では、レンダリング・ユニット150A～Dは、実際のピクセル・データではなく「サンプル」を計算する。これによって、レンダリング・ユニット150A～Dは、1ピクセルにつき複数のサンプルを「スーパーサンプル」または計算することができる。スーパーサンプリングについては、後で詳述する。レンダリング・ユニット150A～Bは、複数のより小さ機能ユニット、たとえば別々のセットアップ／圧縮解除ユニットとライティング・ユニットを含むことができることに留意されたい。

【0051】

C. データ・メモリ

各レンダリング・ユニット150A～Dは、命令およびデータ・メモリ152A～Dに結合することができる。一実施形態では、各データ・メモリ152A～Dは、レンダリング・ユニット150A～Dのためのデータと命令を記憶するように構成することができる。実装態様は様々なものとすることができるが、一実施形態では、各データ・メモリ152A～Dは、2個の8MバイトSDRAMを含み、各レンダリング・ユニット150A～Dに合計16Mバイトの記憶域を与えることができる。他の実施形態では、RDRAM（ラムバスDRAM）を使用して、各レンダリング・ユニットの圧縮解除動作とセットアップ動作をサポートし、SDRAMを使用してレンダリング・ユニット150A～Dの描画機能をサポートすることができる。

【0052】

D. スケジュール・ユニット

レンダリング・ユニット150A～Dとサンプル・メモリ160A～Nとの間にスケジュール・ユニット154を結合することができる。スケジュール・ユニット154は、完成したサンプルを順序づけし、それらをサンプル・メモリ160A～Nに記憶するように構成することができる。より大きな構成では、複数のスケジュール・ユニット154を並列して使用することができることに留意されたい。

【0053】

E. サンプル・メモリ

サンプル・メモリ160A～160Nは、複数のサンプルを記憶するように構成されたスーパーサンプル型サンプル・バッファ162を含む。本明細書では、「スーパーサンプル型サンプル・バッファ」という用語は、サンプルを記憶する1つまたは複数のメモリを指す。前述のように、1つまたは複数のサンプルをフィルタリングして出力ピクセル（たとえば表示装置上に表示されるピクセル）を形成し、記憶されるサンプル数は、単一のフレームをリフレッシュするために表示装置に出力される合計ピクセル数より多くても等しくても少なくともよい。各サンプルは1つまたは複数の出力ピクセルに対応する。本明細書で使用するサンプルは、サンプルの情報がピクセルの最終出力値に影響を及ぼす場合の出力ピクセルに対応する。

【0054】

言い換えると、スーパーサンプル型サンプル・バッファは、複数のサンプルを格納するサンプル・バッファを含む。サンプルは、表示装置上の場所に対応する位置を有する。すなわち、サンプルは、表示装置上のそれぞれの場所にある1つまたは複数の出力ピクセルに影響を与える。これらの場所は、表示装置上のピクセルの中心に対応するか、または表示装置上のピクセル中心の間の位置に対応することができる。記憶サンプル数は、ピクセル場所の数よりも多くすることができる。記憶サンプル数は、ピクセル場所の数よりも多くすることができ、畳込み（フィルタリング）プロセスで複数のサンプルを組み合わせ、表示装置上に表示される1つまたは複数のピクセルを生成することができる。

【0055】

サンプル・メモリ160A～160Nは、複数の異なるタイプのメモリ（たと

例えばSDRAM、SRAM、RDRAM、3DRAM)のいずれでも含むことができる。本明細書に記載の実施形態では、スーパーサンプル型サンプル・バッファを使用するが、他の実施形態では従来のピクセル・フレーム・バッファを使用することもできることに留意されたい。しかし、スーパーサンプル型サンプル・バッファを使用した場合、ピクセル・フィル制限付きであると判断されたグラフィックス・データのセットによって、グラフィックス・システム（または対応するソフトウェア）はレンダリングしているシーン全体のうちの一部分について、サンプル密度を低くし、フレーム・レートを向上させることができる。

【0056】

グラフィックス・プロセッサ90は、特定のサンプル位置決め方式（たとえば規則的グリッド、乱れた規則的グリッドなど）に従って、複数のサンプル位置を生成するように構成することができる。あるいは、サンプル位置は、メモリ（たとえばROMテーブル）から読み取ることもできる。レンダリングするポリゴンを受け取ると、グラフィックス・プロセッサ90は、そのポリゴン内にどのサンプルが入るかを判断する。グラフィックス・プロセッサ90はそれらのサンプルをレンダリングし、次にそれらをサンプル・メモリ160A～Nに記憶する。本明細書では、レンダリングおよび描画という用語は交換可能に使用し、サンプルのカラー値の計算を指すことに留意されたい。このレンダリングまたは描画プロセスでは、アルファ値およびその他の1サンプルについての値も計算することができる。一実施形態では、グラフィックス・プロセッサ90は、前述の計算を行って、レンダリングしているシーンによってフレーム・レートが所定の最低しきい値より下に下がるか否かを判断し、シーンがピクセル・フィル制限付きかポリゴン・オーバーヘッド制限付きかによってレンダリング・パラメータを変えるように構成することができる。

【0057】

E. サンプルーピクセル計算ユニット

サンプル・メモリ160A～NとDAC178A～Bの間に、サンプルーピクセル計算ユニット170A～Dを結合することができる。サンプルーピクセル計算ユニット170A～Dは、サンプル・メモリ160A～Nから選択されたサン

プルを読み取り、次にそれらのサンプルに対して畳込み（たとえばフィルタリング関数および重み付け関数）を実行し、DAC178A～Bに出力される出力ピクセル値を生成するように構成される。サンプルーピクセル計算ユニット170A～Dは、所望の出力のタイプに応じて異なる時点で異なるフィルタリング関数を実行することができるようにプログラムすることができる。一実施形態では、サンプルーピクセル計算ユニット170A～Dは、4×4スーパーサンプル再構築バンドパス・フィルタを実施して、（サンプル・メモリ160A～Nに記憶されている）スーパーサンプル型サンプル・バッファ・データを単一のピクセル値に変換することができる。他の実施形態では、計算ユニット170A～Dは、選択された数のサンプルを平均して出力ピクセルを計算することができる。この平均されたサンプルには、計算するピクセルの中心に近い位置を持つサンプルに多少の重み付けをする変数重み付け係数を乗じることができる。その他のフィルタリング関数も、単独で、またはたとえばテント・フィルタ、円および楕円フィルタ、ミッチェル・フィルタ、バンドパス・フィルタ、同期関数フィルタを組み合わせ使用することができる。

【0058】

サンプルーピクセル計算ユニット170A～Dは、プログラマブル・ビデオ・タイミング・ジェネレータ、プログラマブル・ピクセル・クロック・シンセサイザ、クロスバー機能、およびクロック・ルックアップ・テーブルのうちの1つまたは複数の機能と共に構成することもできる。サンプルーピクセル計算ユニットが各出力ピクセルのタイミングと色を操作した後は、出力ピクセルはDAC178A～Bに送られる。

【0059】

F. DAC

DAC178A～Bは、グラフィックス・システム112の最終出力段階として機能する。DAC178A～Bは、クロス・ユニット174A～Bから受け取ったデジタル・ピクセル・データを、表示装置に送られるアナログ・ビデオ信号に変換する役割を果たす。一実施形態では、アナログ・ビデオ信号の代わりにデジタル・ピクセル・データを出力するために、DAC178A～Bを迂回ま

たはまったく省くこともできることに留意されたい。

【0060】

定義

本明細書では、以下の関数表記を使用する。

s = モデル空間－画面空間スケーリング係数

$f(x)$ = x 以下の面積を有するポリゴンの集合からランダムに選択されたポリゴンの確率

$g(x)$ = ポリゴンの集合によって形成された表面上のランダムに選択された点が x 以下の面積を有するポリゴンに属する確率

$f^{\sim}(x)$ = 単位スケーリング係数 s (すなわち $s = 1$) の場合の $f(x)$

$g^{\sim}(x)$ = 単位スケーリング係数 s (すなわち $s = 1$) の場合の $g(x)$

$p_m(a)$ = 面積 A を有する単一のポリゴンのモデル空間確率分布

α = ビュー・フラスタムの外部にあるポリゴンの部分の推定値

【0061】

ポリゴンのパラメータ化－図6～図11

概念上、モザイク状オブジェクトの3つの一般的クラスがある。すなわち、(1) 特定の対象表面曲率および詳細の基準を満たす事前モザイク化されたオブジェクト、(2) 画面空間におけるサイズ基準を満たすように動的にモザイク化されたオブジェクト、および(3) モデル空間における特定のサイズ基準を満たすように静的にモザイク化されたオブジェクトである。

【0062】

第1のクラスには、パラメトリック表現からハンド・デジタイズされたものであるか、3Dスキャンされて簡略化されたものであるか、またはモザイク化されたものであるかを問わず、ほとんどの従来の三角形に分かたれたオブジェクトを含めることができる。第2のクラスには、マイクロポリゴンを生成するために様々なシェーダによって動的にモザイク化されたパラメトリック・オブジェクトを含めることができる。単純なシェーダには、テクスチャ・マッピング、バンプ・マッピング、およびディスプレイースメント・マッピングが含まれる。パラメトリック表現は、テクスチャ座標を持つポリゴンのように単純なものや、高次NU

RBS（不均一有理Bスプライン）のように複雑なものとしてすることができる。第3のクラスは、いわゆる幾何シェーダであり、一般には、複雑すぎてリアルタイムでは計算することができないシェーダ（たとえばプロシージャ・テクスチャ）を事前適用することによって生じる。これらの異なる各クラスは、ポリゴンの多少異なる画面空間統計を生じさせるが、次の数項で説明する分析ツールは、すべてのクラスに適用される。

【0063】

3Dコンピュータ・グラフィックスにおいてレンダリングされるオブジェクトの一般的な表現は、モデル空間ポリゴンの集合としてである。このようなオブジェクトを、本明細書では幾何形状と呼ぶ。本明細書では説明を簡単にするためにいくつかの例で三角形を使用することがあるが、本開示および特許請求の範囲は、より一般的にすべてのポリゴンに適用可能であることに留意されたい。レンダリング・プロセス中、個々のポリゴンは共通モデル空間に変換され、次に画面空間に射影される。次に、最終レンダリング・プロセスによって、表示装置上に最終的に表示するためにピクセルをフレーム・バッファ内に（またはサンプルをサンプル・バッファに）書き込む。まず、画面空間ポリゴンの特性について説明する。これらの結果は、次に元のモデル空間ポリゴンに一般化される。説明を簡単にするために、この項では、すべてのポリゴンが可視であり、フラスタム、フェース、または隠蔽クリッピングまたはカリング（これらについては後で詳述する）を受けないものと仮定する。

【0064】

画面空間ポリゴンの集合の面積統計値を求めるための1つの手法は、所与の面積の画面ポリゴンの出現頻度のヒストグラムを計算することであろう。合計ポリゴン数によってこのヒストグラムを正規化すると、画面面積 a を有するランダムな画面空間ポリゴンの確率を表す確率分布関数 $p(a)$ が得られる。

【0065】

しかし、これらの確率分布の線プロットは、面プロットの小端に突き当たった極値指数曲線のように見える傾向があるため、視覚的に情報価値がない。興味ある詳細が見えるようにするため、面積の対数軸のようなものを使用してこの確率

分布をプロットすることができる。残念ながら、対数軸の使用によって、確率分布のすぐれた視覚的特性の1つが失われる。すなわち、曲線の下領域がポリゴンの所与の面積の相対母集団を示さなくなる。実際のオブジェクトからの経験的データを使用した場合、確率分布には他の限界がある。数量化効果によって曲線に人為要素が残る可能性があり、それによって平滑化が必要になる。これらの問題を回避するため、累積確率分布を使用することができる。したがって、関数 $f(a)$ を、 a 以下の面積を有するランダムに選択されたポリゴンの確率として定義することができる。 $p(a)$ を仮定すれば、 $f(a)$ は以下のように単に 0 と a の間の $p(a)$ の定積分になる。

【数 6】

$$f(a) = \int_0^a p(x) dx \quad (1)$$

【0066】

ポリゴンの累積面積のための関数、すなわち、レンダリングしているオブジェクトまたはシーンの総表面積がどれだけ、 a 以下の面積のポリゴンによって占められているかの関数を使用することも有用であろう。これを考慮する他の方法は、表面上のランダムに選択した点が a 以下の面積を持つポリゴンに属する確率である。この累積面積確率を、本明細書では $g(a)$ と呼ぶ。 $p(a)$ を仮定すると、 $g(a)$ は以下になる。

【数 7】

$$g(a) = \frac{\int_0^a x \cdot p(x) dx}{\int_0^\infty x \cdot p(x) dx} = \frac{1}{\text{総面積}} \cdot \int_0^a x \cdot p(x) dx \quad (2)$$

【0067】

下段の項は、単にこの幾何形状の合計（画面空間）面積である。これらの定義を使用して、幾何形状の集合について $f()$ および $g()$ を計算することができる、その後でプロットすることができる。

【0068】

次に図6を参照すると、5個の異なる三次元レンダリング・オブジェクトの画像が示されている。これらのオブジェクトは、V22オスプリー航空機250、トリケラトプス252、エンジン254、仏像256、およびティラノザウルス・レックス（Tレックス）の骨格258である。

【0069】

オスプリー250は、約30,000個の三角形を有する視点データベースからの従来のモザイク化オブジェクトである。トリケラトプス252は約250,000個の三角形を有するテクスチャ付きオブジェクトである。これは、約6,000個の三角形を有する視点データベース三次元モデルに画像テクスチャをミップマップするシェーダを適用することによって生成された。エンジン254は、約250,000個、百万の4分の1個の三角形を有するMCADモデルである。仏像256は、約300,000個の三角形を有するCyberwareスキャン・オブジェクトである。Tレックス258は、約130,000個の三角形を有する視点データベース三次元モデルである。

【0070】

上述の $f()$ および $g()$ を使用した、各オブジェクトの $f()$ および $g()$ のグラフを図7ないし図11に示す。これらのグラフでは、 $f(a)$ は黒い実線として図示され、 $g(a)$ は長間隔の破線として図示されている。短間隔の破線として図示されている関数 $e(a)$ と、中間隔の破線として図示されている $h(a)$ は、以下で詳述する関数である。参考のため、図7ないし図11の各グラフの上に三角形数、画面空間面積、および奥行複雑度を示す。各オブジェクトは、個別に描画され、960×680ウィンドウ内に収まるようにスケーリングされている。

【0071】

図7ないし図11のグラフの曲線から様々な観測を行うことができる。まず第一に、 $f^{-1}(0.5)$ が中央値三角形面積であることに注目されたい。第二に、オスプリー250、エンジン254、およびある程度まではTレックス258も、 $f(a)$ 曲線が $g(a)$ 曲線の左に大幅にシフトしている。これは、大部分の

三角形の面積が比較的小さいことと、相当量の面積が、比較的少数の大きな三角形内に閉じ込められていることを示している。三角形の面積の変化はグラフ内で見ることができる。80パーセンタイル・カットの範囲に、一桁と二桁の間の異なるサイズを有する三角形が含まれている。

【0072】

f () および g () の実証的把握

オスプリー 250 は、5 個のオブジェクトの中で最も単純であり、T レックス 258 を除くオブジェクトの 8 分の 1 未満の数の三角形を有する。それにもかかわらず、 $f^{-1}(0.5)$ を調べると、中央値三角形の面積は 2 ピクセル未満であることがわかる。 $g^{-1}(0.1)$ から、面積の 90% が 10 ピクセルを超えるサイズの三角形内に閉じ込められていることがわかる。これらの統計値は、車輪とランディング・ギヤにおける精細な幾何的細部によるものであるとわかる。この種の調査的分析は、累積曲線の使用、すなわち、オブジェクトを分析して、三角形数の減少があるか否か、およびどこにあるかを調べることである。エンジン 254 は、オスプリー 250 の 8 倍の三角形数を有するが、エンジン 254 は他の点では類似した $f()$ 曲線および $g()$ 曲線を有する。その一つの理由は、この 2 つのオブジェクトの画面面積（影響される固有ピクセルではなく、レンダリングされたピクセルの合計数に等しい画面面積）を比較すればわかる。エンジン 254 は、オスプリーの 5 倍の画面面積を有し、したがって、正規化画面面積で表すと、エンジン 254 はレンダリングされたピクセル当たりの三角形の数が 8/5 しか多くない。これを考えると、 $f()$ および $g()$ 統計値が類似するのは意外なことではない。他の 3 個のオブジェクトでは、90% の三角形の面積は 3.5 ピクセル未満であることが明らかである。したがって、これらのオブジェクトは大きな三角形を考慮に入れてモザイク化されていないと思われる。このような多数の小さな三角形は、より少数のより大きな三角形を使用するモザイク化と比較した場合、リアルタイム・ハードウェア・レンダリング機構の負担が一桁大きくなる。曲線の形状は、レンダリングしているオブジェクトのみに依存することに留意されたい。スケールの変更では、グラフ上で曲線が右または左にシフトするに過ぎない。

【0073】

三角形のアスペクト比および歪み統計

レンダリング・パフォーマンスの主要決定要因は三角形の面積であるが、場合によっては他の要因も影響する。したがって、画面空間三角形の完全な特性評価には、三角形の面積だけでなく、三角形のアスペクト比と歪みも含まれる。次に、図12を参照すると、三角形のアスペクト比と歪みの計算を示す図が示されている。本明細書では、アスペクト比は、三角形の高さと三角形の最長稜との比を表す、0と $3^{1/2}/2$ の間の数値として定義する。可能最長アスペクト比は、 $3^{1/2}/2$ （正三角形の場合）である。直角二等辺三角形は、アスペクト比 $1/2$ を有することに留意されたい。アスペクト比は、三角形の幅に多くのピクセルがあるが、高さには1または2ピクセルしかない場合に重要である。これは、ほとんどのハードウェアがこのような三角形内を充填する際に比較的非効率的であるためである。これは、ページ・ミス、バンク断片化、および有効ピクセル探索オーバーヘッドを含む、様々な実施態様固有ボトルネックによるものである。歪みは、第三頂点が三角形の最長稜の最も近接した頂点にどれだけ近接しているかでないと定義され、1（細い三角形）と $1/2$ （対称な三角形）の間で変化する。

【0074】

アスペクト比の実証的理解

図13および図14は、Tレックス258とエンジン254のモデル空間（破線）および画面空間（実線）におけるアスペクト比のヒストグラムを示すグラフである。Tレックスのモデル空間曲線は、 $1/2$ 付近の明白なピークを有し、正方形に近い四辺形からの直角三角形の対が多数あることを示している。仏像256およびテクスチャ化トリケラトプス252（図示せず）のモデル空間曲線も類似しており、 $1/2$ におけるさらに明確なピークを有する。それに対してエンジン254は、大部分0から $1/2$ の範囲にある、はるかに均等な分布を有する。エンジン254も、大部分直角三角形で構築されているが、モザイク化機構のために、それらの多くはより細長い四辺形からのものである。オスプリー250の分布（図示せず）もエンジン254のものに類似している。画面空間への射影によって、アスペクト比確率曲線が左、すなわちより細い三角形の方向にずれる傾

向がある。これは、ほとんどの射影角によって、所与の三角形が一方向に細くなり、細い三角形を太くするのはわずかな射影角のみであるためである。これは、画面空間三角形が細くなる傾向があるという観察（およびその実証的検証）に理論的な基礎を与える。この傾向は、小さく細い三角形にとって大きなパフォーマンス上の不利益があるアーキテクチャには有用である。モデル空間と画面空間の両方の歪みのヒストグラム（図示せず）も類似し、かなり平らになる傾向があった。歪みは一般には、ほとんどのハードウェアのパフォーマンスに影響を与えず、万全を期すために本明細書に記載したに過ぎない。

【0075】

モデル空間－画面空間変換－図15A～15B

次に図15Aを参照すると、モデル空間におけるポリゴン150の例が図示されている。モデル空間とは、レンダリングする三次元オブジェクトを生成するときに使用される座標系を指す。ポリゴン150は、各頂点が (x, y, z) の形式の三次元座標を有する3つの頂点によって決まる三角形である。ポリゴン150がモデル空間から画面空間にどのように変換されるかは、視点152（カメラとも呼ぶ）の相対位置と向きによって決まる。たとえば、視点152が $(100, 50, 0)$ にあり、モデル空間の原点 $(0, 0, 0)$ に向けられている場合、ポリゴン150はモデル空間ビュー（図15Bに示す）から消えるか、または薄い線として現れる。これは、ポリゴン150に深さがなく、視点152がこの稜からまっすぐにポリゴン150を見ている（1枚の紙の縁を見ているのと似ている）ためである。

【0076】

図15Bに、図15Aに示す視点の位置および向きの場合の、画面空間におけるポリゴン150の外観（すなわち、コンピュータの表示装置上で見た場合にポリゴン150がどのように見えるか）を示す。画面空間とは、表示装置上のピクセルの座標系を指す。モデル空間座標と画面空間座標は、レンダリングのために使用される視点の関数として関連している。ポリゴン150は、変換プロセス中に回転およびスケーリングすることができる。このスケーリング・プロセスによって、視点152に近く配置されたオブジェクトは視点152からより離れたオ

プロジェクトよりも大きく見えるようにすることができる。このスケーリング・プロセスは、モデル空間－画面空間スケーリング係数「 s 」として数量化することができる。

【0077】

モデル空間と画面空間の両方のポリゴン統計を計算することができる。モデル空間において統計をとるには、ユーザ・レベルでの幾何形状データの処理を必要とする。これは、1オブジェクトについて1回だけ行えばよい。それに対して、画面空間において統計をとるのはより難しい。レンダリング・パッケージ全体を作成するか、（ソース・コードにアクセスできると仮定して）既存のパッケージに変更を加える。他の問題は、結果がビューに依存することである。計算は難しいが、これらの統計は、画面空間におけるレンダリングされたポリゴンの振舞いを理解するのに有用である。

【0078】

しかし、モデル空間統計値の変換によって所与の幾何形状の平均画面空間統計値をモデリングすることが可能である。複数の異なる視角にわたって画面空間統計値の平均をとることによって、限界内で、結果は単一のポリゴンをすべての可能な角度で射影する統計値を使用したモデル空間統計値の畳込みのように見えるはずである。さらに、高ポリゴン数オブジェクトの場合、幾何細部があらゆる向きに存在する傾向があるため、画面空間統計値は向きによって大幅には変化しない傾向がある。

【0079】

たとえば、モデル空間における面積 A の単一のポリゴンを考えてみる。これは、ディラック・デルタ関数 $p_m(a) = \delta(A - a)$ のモデル空間確率分布を有する。ポリゴンのスケールが視点からのポリゴンの距離に対して小さいものとする、ビュー射影の効果は以下の要素に因数分解することができる。すなわち、（1）モデル空間－画面空間スケーリング係数「 s 」（最大画面空間面積 sA になる）と、（2）ポリゴンの側面での回転である。したがって、このポリゴンを複数の異なる視点の向きにわたって（ただしすべて同じ距離から）射影する場合、これらの関数を、まず s によってスケーリングし、次に画面空間にスミア（ sm

ear) する。したがって、画面空間統計値は以下のようになる。

【数 8】

$$p(a) = \begin{cases} \frac{1}{sA} & 0 \leq a \leq sA \\ 0 & \text{上記以外の場合} \end{cases} \quad (3)$$

これは、射影されたポリゴンの条件付き分布が均一であることを意味する。この証明の概要を以下に示す（「射影の不変性の証明」という見出しの項を参照）。

【数 9】

$$f(a) = \begin{cases} \frac{a}{sA} & 0 \leq a \leq sA \\ 0 & \text{上記以外の場合} \end{cases} \quad (4)$$

【数 10】

$$g(a) = \begin{cases} \frac{a^2}{(sA)^2} & 0 \leq a \leq sA \\ 0 & \text{上記以外の場合} \end{cases} \quad (5)$$

【0080】

一般的な場合、モデル空間確率分布を $p_m(a)$ 、モデル空間—画面空間スケール係数を s とすると、総画面空間統計（すなわち単位スケール係数 $s=1$ の場合）は、以下の数式によって求められる。

【数 11】

$$\hat{f}(a) = \int_0^a p_m(x) dx + \int_a^\infty \frac{a}{x} \cdot p_m(x) dx,$$

ただし $f(a) = \hat{f}(as)$ (6)

【数 12】

$$\hat{g}(a) = \int_0^{\infty} \int_0^{\min(y, a)} 2 \cdot \frac{x^2}{y^2} \cdot p_m(y) dx dy,$$

$$\text{ただし } g(a) = \hat{g}(as) \quad (7)$$

【0081】

これらの関数では、 s の変化の結果、一定した曲線が a を基準にしてシフトするに過ぎないことが暗黙に示されている。この推測は、 $f(a)$ および $g(a)$ の画面空間プロットを、同じオブジェクトについて $p_m()$ トレースを数値積分することによって得られるプロットと比較することによって実証的に確認された。

【0082】

図16を参照すると、Tレックス258のランダムな視点の向きの場合の、 $f(a)$ の実証的プロットを示すグラフが示されている。この図では、細い各黒線が、単一のランダムな視点の向きの場合の $f(a)$ の実証的プロットを表し、太い黒線は、 $f_m(a)$ のモデル統計を数値的に積分して得られた $f(a)$ である。予測は、曲線の形状と曲線の位置の両方についてであった。位置は、動的モデリングとビュー変換によって計算された s の値を使用して生成した。視角による多少の変動があるが、全体的な予測はかなりよく合っている。同様の手法を使用してアスペクト比に与える射影の影響を予測することができる。

【0083】

レンダリング・パフォーマンスのモデリング

機械のレンダリング・パフォーマンスの単純化されたモデルは、ピクセル数／秒の単位で測定された、単なるピクセル・フィル・レートである。このモデルでは、所与の幾何形状の1フレーム当たりのレンダリング時間は、単に、レンダリングされた合計画面空間面積をピクセル・フィル・レートで割った値になる。レンダリングしているシーンが主として大きなポリゴンから成る場合、これはかなり正確なモデルである。しかし、前述のように、大きなポリゴンはめったにない。

【0084】

レンダリング・パフォーマンスのより現実的なモデルは、どのような所与のレンダリング・ハードウェアおよびレンダリング属性の設定の場合でも、それを下回るとレンダリング・ハードウェアがフィル制限付きではなく最小ポリゴン処理オーバーヘッド時間制限付きとなる、ポリゴンの臨界面面積 a_c が存在することを考慮に入れる。光源の数または複雑さの増大など、1頂点ごとのレンダリング属性の変化は、最小ポリゴン処理オーバーヘッド時間を増大させる傾向がある。透明性の使用可能化、テクスチャリング、アンチエイリアシング、またはテクスチャリングの複雑さの増大などのピクセルごとの属性の変化は、1ピクセルごとのフィル・レートを低下させる傾向がある。

【0085】

前述のように図2は`glperf` (OpenGLプログラム用のパフォーマンス・タイミング・アプリケーション・プログラム) を使用した実際のハードウェアのタイミングによって得られた実証的データに基づく、ポリゴン・レンダリング時間とポリゴン面積との関係のプロットである。水平線 (データに合わせた) は、最小ポリゴン処理オーバーヘッド時間限界を表す。傾斜した線は、毎秒2億2千4百万ピクセルの漸近的フィル・レート限界を表す。このプロットを生成するために使用した特定の機械の場合、レンダリング属性の特定のセット a_c は約38ピクセルであり、予測レンダリング・パフォーマンス・モデルとかなりよく合致している。

【0086】

ハードウェアのレンダリング・パフォーマンスを単一の数値によって特性評価するには、「疑似面積」という概念を使用することができる。疑似面積によって、最小ポリゴン・オーバーヘッドがそれに相当する面積に変換される。この概念、ハードウェア固有臨界面積 a_c 未満の面積を有するポリゴンはいずれも、疑似面積 a_c を有すると言えるというものである。 a_c より大きいポリゴンは、「実面積」のみを有すると言える。ポリゴンの「有効面積」は、その面積が a_c のどちら側にあるかによって、その疑似面積または実面積であると定義される。「通常面積」という用語は、面積の標準の意味を示す。これらの用語を拡大して、構成ポリゴン (またはポリゴン) の個々の面積を合計することによって幾何形状全体

に適用することができる。したがって、所与の幾何形状の1フレーム当たりのレンダリング時間は、幾何形状の有効面積をグラフィックス・システムのピクセル・フィル・レートで割った値である。幾何形状についてこれらすべての面積を、 a_c によってパラメータ化することによって、アーキテクチャとは独立した方式で特性評価することができる。形式的には、画面空間ポリゴン確率 $p()$ を持つ（すべてが正面を向いている）ポリゴンからなる所与の幾何形状について、 $a_c = a$ の場合、これらの用語は以下の数式で定義される。

【数13】

$$\text{通常面積}(a_c) = \int_0^\infty x \cdot p(x) dx = \text{総面積} \quad (8)$$

【数14】

$$\text{疑似面積}(a_c) = a_c \cdot n \cdot \int_0^{a_c} p(x) dx = a_c \cdot n \cdot f(a_c) \quad (9)$$

【数15】

$$\text{実面積}(a_c) = \int_{a_c}^\infty x \cdot p(x) dx = \text{総面積} \cdot (1 - g(a_c)) \quad (10)$$

【数16】

$$\begin{aligned} \text{有効面積}(a_c) &= \text{疑似面積}(a_c) + \text{実面積}(a_c) \\ &= a_c \cdot n \cdot \int_0^{a_c} p(x) dx + \int_{a_c}^\infty x \cdot p(x) dx \end{aligned} \quad (11)$$

【0087】

幾何形状のレンダリング・パフォーマンスは、関数 $e(a)$ を、値 $a_c = a$ の場合の幾何形状の有効面積とその通常面積との比であると定義することによってアーキテクチャとは独立した方式で特性評価することができる。関数 $e(a)$ は、以下のように事前定義された関数で表して定義することができる。

【数17】

$$e(a_c) = \frac{\text{有効面積}(a_c)}{\text{通常面積}(a_c)} = \frac{a_c}{\text{平均面積}} \cdot (f(a_c) + 1 - g(a_c)) \quad (12)$$

【0088】

関数 $e(a)$ は、幾何形状がレンダリングするフィル制限付きの場合よりも遅い量である (値 $a_c = a$ の場合)。

【0089】

しかし、これは必ずしも完全なモデルではない。最近の複雑な幾何形状は典型的には、背面が一般には見えないことを意図した閉殻である。したがって、一般には背面カリングを行う (本明細書の5つの幾何形状例の場合はこれであった)。パフォーマンスのより正確な推定のために、これらの背面向きポリゴンも考慮に入れることができる。多くの機械では、背面向きポリゴンは正面向きポリゴンの最低処理時間よりも短い時間でなくすことができない。したがって、背面向きポリゴンは疑似面積の a_c を有しているといえる。上掲の式を拡張して背面向きポリゴンを含めるために、平均してポリゴンの半分が背面向きになるものと仮定する。この結果、疑似面積 a_c を持つ追加の「 n 」個のポリゴンができる。したがって、疑似面積および $e(a)$ のより完全な定義を以下に記載する。

【数18】

$$e(a_c) = \frac{a_c}{\text{平均面積}} \cdot (f(a_c) + 1) - g(a_c) \quad (13)$$

【数19】

$$e(a_c) = \frac{a_c}{\text{平均面積}} \cdot (f(a_c) + 1) - g(a_c) \quad (14)$$

【0090】

上記の数式で、 n はやはり正面向きポリゴンの数であり、この幾何形状内のポリゴンの総数ではない。機械によっては、背面向きポリゴンを、最小レンダリング・ポリゴンよりも少ないオーバーヘッドでなくすことができるものもある。そのような場合、背面向きポリゴンは a_c よりも小さい疑似面積を有することがある。この場合、疑似面積に加えられる $a \cdot n$ 項を、背面処理時間と最小ポリゴン・レンダリング時間との実際の比に n を掛けた項に置き換えることができる。

【0091】

次に図7ないし図11に戻ると、上記の定義した $e(a)$ は短間隔の黒い破線でプロットされている。プロットされている他のすべての関数とは異なり、 $e(a)$

a) は0-1の縦軸ではなく、0-10Xの拡大された範囲(グラフの右側に示されている目盛り)を有することに留意されたい。疑似面積効果について考察する関連のある方法は、疑似面積と $a_c = a$ の純粹に真の面積との比をプロットすることである。これが $h(a)$ であり、図の中間隔の破線で示されている。関数 $h(a)$ は、0から1まで変化し、所与の幾何形状が $a_c = a$ のグラフィックス・システムにおいて変換限界とフィル限界とに費やす時間の割合としてそのまま読み取ることができる。

【数20】

$$h(a_c) = \frac{\text{疑似面積}(a_c)}{\text{有効面積}}$$

$$= \frac{a_c \cdot (f(a_c) + 1)}{a_c \cdot (f(a_c) + 1) + \text{平均面積} \cdot (1 - g(a_c))} \quad (15)$$

【0092】

レンダリングしている幾何形状のモデル空間から画面空間のスケールが拡大または縮小した場合、 $e(a)$ および $h(a)$ の曲線は右または左にシフトすることに留意されたい。しかし、所与の機械および所与のレンダリング・パラメータのセットについて、 a_c は典型的には定数であり、したがって幾何形状についての機械のレンダリング効率は、その幾何形状のスケールが変化すると変化する。

【0093】

$e()$ および $h()$ の実証的理解

図7において、その面積の90%はサイズが10ピクセルを超える三角形内にあるが、オスプリー250の $e(10)$ のグラフは、臨界面積 a_c が10の機械がオスプリー250をフィル制限レート(速度の半分以下)でレンダリングすることになることを示している。これは、三角形の75%が10ピクセル未満の面積であるためである。臨界面積 a_c が38ピクセルの図2の機械は、フィル制限付きより6倍遅く(このオブジェクトに関する実証的数値は6.8倍遅く、予測値は6.4倍であった)。 a_c が1ピクセルの機械であっても、Tレックス258、仏像256、およびトリケラトプス252の場合のフィル・レート制限の約2倍近く遅くなる。その理由は、これらの薄暗いオブジェクトの $g(1)$ からわか

る。すなわち、レンダリング時間の50%~70%が疑似領域に拘束されている。

【0094】

ハードウェア設計への $e()$ および $h()$ の適用

これらの関数は、グラフィックス・ハードウェア設計に適用することができる。レンダリングする対象幾何形状の集合がある場合、ポリゴン・レートの漸進的向上とフィル・レートの漸進的向上との差の兼ね合いを直接図ることができる。フィル・レートが2倍に向上すると実面積のレンダリング時間が2分の1になることができるが、 a_c も2倍になる。全体的な効果は、合計レンダリング時間を短縮することであるが、幾何形状がすでに90%疑似面積制限であった場合($h(a)$ を調べることによってわかる)、フィル・レートの2倍の向上によって、レンダリング時間の向上は10%未満になる(実面積の一部が疑似面積になる)。幾何形状の疑似面積制限が50%に過ぎない場合であっても、レンダリング・レートを無限に向上させても、レンダリング時間の向上は2倍にしかない。基本ポリゴン・レートを2倍にすると、 a_c が2分の1になる。幾何形状が90%疑似面積制限であった場合、レンダリング時間は45%しか向上しない(疑似面積の一部が実面積になる)。差益は a_c 付近の曲線の勾配に依存する。

【0095】

たとえば、仏像オブジェクトの $h()$ は a_c が2で90%であり、 $f(2)$ はフィル制限付きより4倍遅い。 a_c を1に変更すると $h()$ が70%に減少し、 $e()$ が約2.3倍になり、レンダリングが1.7倍速くなる。それに対して、フィル・レートを2倍にした場合、 a_c は2から4に倍増し、 $e()$ は4倍から7.8倍にほぼ倍増して、フィル・レートの2倍の利得をほぼ完全に帳消しにする。

【0096】

汎用3Dレンダリング・ハードウェアは、あらゆる種類のオブジェクトのレンダリングを加速し、ほとんどのオブジェクトに対して無視できるほどの影響しか与えないフィル・レートの向上でも、ある種のオブジェクトにとっては有効になる。しかし、全体としては、ターゲット市場のために、レンダリングするオブジ

エクトの統計値を測定することができ、バランスのよいハードウェア・アーキテクチャによってポリゴン・レート・ハードウェア資源とフィル・レート・ハードウェア資源との兼ね合いを図って、それらのオブジェクトのレンダリング時間を最小限にすることができる。これは一般には、 $e(a_c)$ を小さい整数範囲に維持することによって実現することができる。

【0097】

より複雑なモデル

臨界面積 a_c 付近では、フレーム・バッファ・メモリの断片化、不十分な頂点チェーニング、および低アスペクト比ポリゴンの影響のため、ハードウェアは多少低いフィル・レートを有することがある。必要であれば、特定のレンダリング・ハードウェアが与えられた場合、 $a = a_c$ における上記で概説した関数のより正確な機械固有値を計算することができる。このより正確な値は、上記およびその他の影響を考慮に入れることができる。

【0098】

上記で開示した汎用モデルの他の限界は、モデル空間におけるポリゴンの向きの等方性分布を前提としていることである。これは、常にそうであるとは限らない。たとえば、図16のシーンでTレックス258における多数の平たい肋骨は、図16で見られる変化を生じさせている。このような幾何形状の振舞いは、少数のビュー固有統計値の適切な補間によってよりよく近似計算することができる。これらの限界にもかかわらず、(14) および (15) に示した関数 $e(a)$ および $h(a)$ は、所与の幾何形状の潜在的レンダリング・パフォーマンスを把握するための、アーキテクチャとは独立したすぐれた方法となる。次の項では、レンダリング時間予測関数を定義し、ランタイム負荷バランスを導くためにこの関数をどのように使用することができるかを示す。

【0099】

レンダリング制御への応用

リアルタイム・シミュレーションの応用分野では、重要な特徴は予測可能で一貫したフレーム・レンダリング時間である。一貫したフレーム・レートを実現するのに役立つ従来の技法は、詳細レベル (LOD) オブジェクトである。LOD

オブジェクトの使用には、ポリゴン数別に分類されたオブジェクトのいくつかの代替幾何学表現を記憶する必要がある。フレームごとのレンダリング時間が所定の最低しきい値に近づくかまたはそれを下回ると、LODオブジェクトの現行表現を変更して、よりポリゴン数の少ないものを使用することができる。あるいは、ビュー依存モザイクをオンザフライで生成することができる。これらの技法は、オブジェクトがポリゴン処理オーバーヘッド制限付きである場合には機能するが、オブジェクトがフィル・レート制限付きである場合には役に立たない。

【0100】

オブジェクトがフィル・レート制限付きである場合、オブジェクトを完全に削除することができる。あるいは、グラフィックス・システムを、所与のオブジェクトのピクセル面積をフレームごとに減らすように構成することができる。他の代替策は、表示解像度を動的に低下させ、それによってそのシーン全体のピクセル数を減らすことであろう。他の実施形態では、選択された背景オブジェクトを削除することができる。

【0101】

スーパーサンプリングを使用するアーキテクチャの場合、フレーム・レートが下がるにつれてオブジェクトまたはシーン全体のサンプル数またはサンプル密度を下げるることができる。関数 $h(a)$ は、所与のオブジェクトがフィル制限またはオーバーヘッド制限である程度を判断し、さらにそれがLODオブジェクト内の他の選択肢の場合にどれだけ変化するかを判断する形式的方法を与える。これによって、フレーム・レート制御についてよりグローバルで正確な決定を行うことができるので有利である。

【0102】

幾何形状レンダリング時間のリアルタイム予測

関数 f^* および g^* を使用して、所与の幾何形状をレンダリングするのにどれだけの時間がかかるかを予測する正確なリアルタイム・アルゴリズムを定義することができる。オフライン処理の一部として、幾何形状の $p_m()$ を計算し、そこから数値積分によって f^* および g^* を計算することができる。図2を生成したような手続きを使用して、別個のオフライン・プロセスにより、(所与のハー

ドウェア・アーキテクチャのための) レンダリング属性の重要なセットについて a_c のいくつかの値を計算することができる。次に、ランタイムで、モデリング・マトリックスおよびビューイング・マトリックスからスケーリング係数 s を計算することができ、以下の数式によって幾何形状のレンダリング時間を推定することができる。

【数21】

$$\begin{aligned} \text{レンダリング時間} &= \text{有効面積} / \text{ピクセル・フィル・レート} \\ &= \frac{(a_c \cdot n \cdot \hat{f}(a_c \cdot s) + 1) + \left(\frac{1}{2} \cdot s \cdot \text{合計モデル面積} \right) \cdot (1 - \hat{g}(a_c \cdot s))}{\text{ピクセル・フィル・レート}} \end{aligned} \quad (16)$$

【0103】

事前計算した合計モデル空間面積に2分の1の s を乗じることによって、数式(2)からの幾何形状の総画面面積を推定することができる。これを数式(16)で使用した。

【0104】

レンダリング時間は、1つまたは複数のオブジェクト変形体の複数の異なる視点についても計算することができる。次に、計算されたレンダリング時間を平均して、各オブジェクト変形体の総推定レンダリング時間を求めることができる。グラフィックス・システムは、次に、この平均総推定レンダリング時間に基づいて、所望の最低フレーム・レートと整合した視覚的に最も現実感のあるオブジェクト変形体を選択することができる。他の実施形態では、オブジェクト変形体の可能なすべての視点についてレンダリング時間を計算することができるか、または、選択された代表的な視点(たとえば、オブジェクトまたは発生する可能性が最も高い事前定義ビューの、重なり合った対称部分または重なり合わない対称部分)のみを使用することができる。他の実施形態では、畳込み関数を使用して、オブジェクトの可能なすべての視点についてレンダリング時間を計算し、平均することができる。

【0105】

フラスタム・クリッピング

ある種の実施形態では、ビュー・フラスタム・カリングの効果を考慮にいれるようにレンダリング時間予測の方法を構成することができる。ビュー・フラスタム・カリングとは、表示装置の現行可視領域内にないポリゴンを廃棄することを指す。ほとんどの最新の機械では、ビュー・フラスタム外のポリゴンは、背面カリングと同様の処理時間中に単純に拒絶される。すなわち、通常、これらのポリゴンは疑似面積 a_0 を有することになる。1つまたは複数の断片に実際にクリッピングされたポリゴンは、一般にかなり長時間であるがそれに対応したレートを要し、その影響は通常無視することができる。ある種の実施形態では、幾何形状のうちのビュー・フラスタム外にある部分の推定をランタイムに行うことができる。この部分を α とすると、書き換えられたレンダリング時間予測関数は以下のようなになる。

【数 22】

$$\text{レンダリング時間} = \frac{(1 - \alpha) \cdot \text{有効面積} + \alpha \cdot a_0}{\text{ピクセル・フィル・レート}} \quad (18)$$

【0106】

レンダリング品質制御

他のある種の応用分野では、制御定数がフレーム・レートではなく画像品質である場合がある。レンダリングされた画像品質は、表示システム、物理的な見る人の知覚能力、および画像の内容の組合せのナイキスト・レートを下回る多数のポリゴンのサイズの維持に関連するため、上記の曲線は画像品質を制御する形式的方法ともなる。具体的には、ユーザは、(中央値面積サブピクセルを維持するために) 現行の s について $f(1)$ が 0.5 以下になるように詳細レベル・オブジェクトを選択したい場合がある。しかし、一般に、このしきい値は、定性的判断の選択肢であり、本明細書に示す例示のオブジェクトのほとんどを含めて、多くの幾何形状の場合、中央値面積を 2 と選定しても、知覚可能な品質はほとんど失われない。その理由は、視覚的に興味深い高空間周波数は、そのような分布におけるサブピクセル・サイズの、数の相当ある少数ポリゴン内にある傾向があるためである。ポリゴンの中には 1 ピクセルよりも大きいものがあるということとは

、表示装置のナイキスト・レートを妨げず、そのようなポリゴンは単に、ほとんどのオブジェクトが持つ、より低い空間周波数領域を示すに過ぎない。これは、2D画像圧縮技法に、視覚的に知覚可能な人為構造が少ないほとんどの画像のすべての領域において高周波数エネルギーをうまくエンコードさせない統計的引数に類似している。面積がわずかなピクセル数しかないポリゴンの場合、グーロー・シェーディングのほとんどの人為構造は消えることにも留意されたい。多くの高品質ソフトウェア・レンダリング・パッケージは、ポリゴンのサイズが1ピクセルに近づくと、単純なフラット・シェーディングを使用する。

【0107】

可変分解能表示装置への拡大

従来、詳細レベル・オブジェクトの選択決定は、サイズおよびオブジェクト重要度情報に基づいて時間的に行われている。しかし、可変分解能サンプル・バッファでは、LODの決定は、画面のうちの幾何形状をレンダリングする領域内にある最小局所ピクセル・サイズも考慮に入れることができる。上記で概説した方法では、これは、この局所ピクセルサイズに適合するようにsを適切に事前スケールリングすることによって処理される。

【0108】

この実証的結果、すなわち、中央値ポリゴン・サイズが1ピクセルに近づいた後はほとんどの幾何形状は画像品質が向上しないということは、レンダリングの最終的パフォーマンス要件を理解する上で役立つ。所与の表示装置の適切な最大ポリゴン・レンダリング・レート目標を推定する方法を可能にするのは、この1ピクセル当たり1以下のポリゴン密度数値が人間の視覚系の限界および物理表示装置の限界と組合わさった場合である。

【0109】

人間の視覚の限界

すべての3Dレンダリングの最終消費者は、人間の視覚系である。表示技法とリアルタイム・ハードウェア・レンダリング速度が向上し続けるに伴い、グラフィックス・システムは人間の視覚系の入力能力をまさに超えようとしている。単一ユーザと60Hzの持続レンダリング・フレーム・レートを持つ機械では、現

在のCRTでさえも、中心窩が見ている場所から離れた領域における人間の視覚系の最大空間周波数検出能力を超えている。中心窩は、人間の網膜のうちでも最も鋭敏な視覚を持つ領域である。

【0110】

この状況を利用するために、ハードウェア・レンダリング・アーキテクチャは、何らかの形態の可変分解能サンプル・バッファを実施することができる。このようなサンプル・バッファでは、空間的分解能は固定せず、人間の視覚の可変分解能性質に適合するように（たとえばフレームごとに）プログラムすることができる。このようなピクセルはアンチエイリアシングすることができ、アンチエイリアシング・フィルタの周波数遮断も、局所有効ピクセル密度に合わせて動的に変化するように構成することができる。

【0111】

最高分解能知覚可能ピクセル：28アーク秒

人間の目によって知覚可能な最高空間周波数は、いくつかの物理的要因によって制限される。瞳の回折限界、中心窩錐体間隔、神経トレースおよび生理学検査すべてによって、1アーク秒当たり約1サイクル（半アーク秒ピクセル）の最大知覚周波数が確認される。これは、100%コントラストを含む最適（ただし非バーニア）条件下にある。直接比較できるわけではないが、いわゆる「正常視力（20/20 vision）」は2倍の画像特徴検出を示す。

【0112】

バーニア条件は、たとえば大きな視覚オブジェクトの角位置においてわずか3アーク秒のシフトを検出することができる場合の超鋭敏の一般的な例である。この場合、視覚系は多数のより低周波のサンプルからより高い空間周波数情報を再構築している。しかし、視覚系は、アンチエイリアシング・プロセス中により高い空間周波数が存在する限りは、より低周波数のレンダリングされた3Dグラフィックス画像についても同じことを行うことができる。

【0113】

可変分解能：1/2@±1度、1/4@±2度、1/8@±5度、1/16@±12度—図17A～17B

しかし、この高解像度は、視野の中心2度にも適用される。この領域外では、光学的限界よりも高速であっても、錐体間隔と測定された知覚の鋭敏さは低下する。多くの参考書では、この低下は鋭い尖点としてプロットされている。しかし、この表現では、視野の高空間周波数知覚領域がどれほど小さいかが適切に示されない。図17Aに、このデータを単位球面上の代替視覚化表現として、すなわち、この4πステラジアン視野のどの部分がどのような分解能で知覚されるかをプロットする。5つの疑似カラー・バンドがあり、各カラー・バンドは2分の1のパーセプトロン分解能に対応する。図17bは、この球面の中央領域の拡大図である。最中心領域は、中心窩の中心±1度に対応する。そこから±2度までの第2の領域、±5度までの第3の領域、±12度までの第4の領域、および人間の顔による光学縁までの第5の領域がある。白は不可視領域を表す。この光学縁は、複雑な形状をしており、個人においても文献においても異なる。これらの計算のために、Scott GrigsbyおよびB. Tsouによる「Visual Processing and Partial-Overlap Head Mounted Displays」という論文(Journal of the Society for Information Display, 2. 2 (1994年)、69~74ページ)からのデータを使用した。このデータは、水平方向に-59度~+110度、垂直方向の-70度~+56度までに変化する最大視野を有する。この180度を超える視野の両側を示すために、1つは右目、1つは対称に逆転した左目のための、2つの単位球面を示す。したがって、凝視の方向がわかっている場合、視野全体にわたって、人間の視覚系は、中心窩分解能が視野全体に使用可能であるとすれば認識可能であるはずの視覚的細部の約15分の1しか知覚することができない。

【0114】

これが3Dグラフィックス・システムに与える可能性のある影響を把握するため、図18にいくつかの代表的な表示装置の推定視覚および表示パラメータの比較を示す表を示す。この表で、列400は様々な表示装置を示す。矩形の表示装置は、その対角線寸法と典型的なユーザ目視距離によって特徴づけられている。最下部の2つの項目は、視覚系と非追跡視覚系(全球面)の純粹限界である。列

402は、表示装置のピクセル解像度を表す。映画解像度は、35mm製品フィルムの実証的数値である。これらの数値は、装置のアスペクト比も決定する。列404は、表示装置のピクセル・サイズである。これは、アークの分単位で表した単一表示ピクセルの角サイズである。列406は、ステラジアン単位で表した表示装置の総立体角視野（FOV）を示す。列408は、均一な28アーク秒知覚を仮定した視野内の最大人間知覚可能ピクセル数を示す。これは単に、列406のステラジアン内に収まる、列404のサイズのピクセルの数である。列410は、前の列と同じ情報を示すが、より実地的な1.5アーク分知覚ピクセルの場合のものである。列412は、図17A～図17Bの可変分解能知覚を仮定した視野内の最大人間知覚可能ピクセル数を示す。列414は、表示装置自体のピクセル限界（列402の数値の乗算）を示す。列416は、表示装置限界と目の限界の両方を考慮に入れた知覚可能ピクセル数を示す。これは、目または表示装置の限界であった表示FOV内の各領域を調べ、いずれか少ない方のみをカウントすることによって計算した。列418は、前の列の限界を、以下で詳述する追加のモデルを使用した最大ポリゴン・レンダリング・レート（毎秒10億ポリゴン単位）として示す。

【0115】

図18の数値の多くを計算するために、単位球面を、各小部分がそれ自体の局所最大知覚可能空間周波数を持つ216個の小部分に分割した。次に、これらの部分と表示FOV縁（または全眼の場合は視野の縁）との交点に対して積分を行う。物理的に平坦な表示装置上の一様なピクセルの角サイズは一定ではない。すなわち、ピクセルは軸から離れると小さくなる。この効果はほとんどの表示装置では小さいが、視野がきわめて広い表示装置ではかなり大きくなる。しかし、実際の表示システムは、複数の表示装置および／または光学系を使用してこの問題に対処しているため、表中の数値では、簡単にするためにこの効果は考慮に入れなかった。

【0116】

この表にはいくつかの注意項目がある。人間の片目のFOVは、4 π ステラジアンFOVの約3分の1である。ワイドスクリーン映画は、この目のFOVの1

2分の1に過ぎず、通常のテレビジョンは百分の1以下である。非追跡（所定位置における）回転観察者の周りあると仮定した球面表示装置は、全視覚系忠実度を生じさせるために、（片目について）フレームごとに10億ピクセルの3分の2以上をレンダリングし、表示する必要があることになる。

【0117】

レンダリングの限界

以下の単純なモデルによって、リアルタイム・システムに必要と考えられる最大レンダリング・レートの1推定値を示す。

$$\Delta/\text{秒} = \text{フレーム} \cdot \text{レート} \cdot (\text{目の数})$$

$$\cdot \text{スクリーン} \cdot \text{ピクセル数} \cdot \text{デプス複雑度} \cdot \Delta/\text{ピクセル} \quad (18)$$

この最後の項の実証的推定値は約1である。前記では、表示装置と視覚に基づく画面ピクセル数の推定値について説明した。60Hz以上という想定以外には、フレーム・レートについては詳述しなかった。急速に変化する複雑なレンダリング画像と人間の視覚系との相互作用についてはあまり知られていない。現在のところ良い手法は、十分に高いと推定される値を選ぶことである。（300Hzを超える）きわめて高いレンダリング・フレーム・レートは、従来のコンピュータ・グラフィックス技法よりも人間の視覚系とより自然に相互作用して、動きぼやけ効果を生じさせると推測する者もいる。

【0118】

表中のレンダリングされたピクセルは、スーパーサンプルまたはエリア・カバレッジ技法に基づいて、高品質再サンプリング・フィルタを使用してアンチエイリアシングすることを仮定したものである。表中のピクセル数にスーパーサンプリング密度を乗じて、ピクセル数ではなくサンプル数を求めることができる。ポリゴン統計値は、デプス複雑度にはわずかしき影響を与えない。しかし、妥当な隠蔽カリング技法を仮定すると、現在の経験では、多くの実施形態におけるデプス複雑度は、ほとんどの（ただし決してすべてではない）場合、3から6の範囲に維持することができる。例示のために、デプス複雑度6を仮定する。これらの仮定を使用し、数式（18）を使用して（立体表示装置の場合は両目を使用）表の列418を計算した。これらの数値の単位は毎秒10億ポリゴンである。これ

らの仮定の下に、人間の視覚系を飽和させるのに必要な最大ポリゴン・レンダリング・レートは、以下ようになる。

60Hz・2つの目・14.78Mピクセル・6DC

・1Δ/ピクセル=10.64B Δ/秒 (19)

【0119】

これは、毎秒100億をわずかに超える。ほとんどの従来の表示装置では、飽和値は、毎秒5億ポリゴン以下である。ここに示す数値は、理論上の最小計算値でも最大計算値でもない。これらは、控えめな「平均的な場合」の推定値であり、これらの前提条件のいずれの変更も結果に大きな影響を及ぼす可能性がある。

【0120】

画面空間ポリゴンの累積統計の対数プロットが、ポリゴンの実証的振舞いを理解するための有用な洞察を与える。この情報は、リアルタイム・レンダリング・ハードウェアの設計およびリアルタイムでレンダリングする三角形に分かたれたオブジェクトの設計において兼ね合いをはかる際に使用することができる。これらの画面空間統計値をどのようにしてモデル空間統計値から直接計算することができるかを示した。これらと同じ関数を使用し、疑似面積の概念を使用してアーキテクチャとは独立した方式でレンダリング・パフォーマンス関数を定義することができる。これらのパフォーマンス関数を高度にしたものを、フレーム・レートの管理またはリアルタイム・レンダリングの質の制御に使用することができる。最低に近い視覚品質は、中央値ポリゴンのサイズが1ピクセル近くに留まることを必要とする。

【0121】

次に図19Aを参照すると、三次元グラフィックス・データのセットのレンダリング速度を効率的に計算する方法の一実施形態が示されている。この実施形態では、上記で概説した数式から得られるいくつかの値を事前計算する。このフローチャートでは、これらの計算が順次に行われているように示されているが、これらの計算は、並行して、または図に示されている順序とは異なる順序で行うこともできることに留意されたい。まず、 a_c 、 $p_m(x)$ 、 $f^-(a_c s)$ 、 $g^-(a_c s)$ 、 s 、および合計モデル空間を計算する（ステップ300～308）

。次に、ビュー・フラスタム・カリングを考慮に入れる場合（ステップ300）、リアルタイムまたはニア・リアルタイムで α を計算する（ステップ314）。次に、レンダリング時間を、有効面積をピクセル・フィル・レートで割った値として計算する（ステップ312）。任意選択ステップ314と同様に、ステップ312もリアルタイムまたはニア・リアルタイムで行って、レンダリング時間のフレームごとの計算を可能にすることができる。

【0122】

次に、計算されたレンダリング時間を所望のフレーム・レートと比較する（ステップ320）。計算されたレンダリング時間が、所定の最低フレーム・レートを満たすのに十分な速さである場合、グラフィックス・システムはフレームを現行パラメータを使用してレンダリングすることができる（ステップ322）。計算されたレンダリング時間が遅すぎて所望のフレーム・レートを満たさない場合、数式（15）を使用して、フィル・レート制限とポリゴン・オーバーヘッド制限のいずれであるかを判断する（ステップ326）。フレームがポリゴン制限の場合、グラフィックス・システムは、ポリゴンの数を減らすようにレンダリング・パラメータを修正する（ステップ330）。前述のように、これはいくつかの方法で（たとえば、より少数のポリゴンを使用するLODを選択したり、オブジェクトを動的により少数のポリゴンにモザイク化したりすることによって）行うことができる。フレームがピクセル・フィル制限である場合、グラフィックス・システムは、ピクセル数またはサンプルを減らすようにレンダリング・パラメータを修正することができる（ステップ332）。前述のように、これも、ピクセルあたりの計算サンプル数を変更する（スーパーサンプル型システムにおいて）したり、レンダリングしているオブジェクトまたはフレームのサイズを動的に変更することを含めて、いくつかの異なる方法で行うことができる。他の代替方法は、特定の背景オブジェクト（たとえばそのフレームを生成したソフトウェア・アプリケーションによってより重要でないと指定されているオブジェクト）を廃棄することである。

【0123】

この方法は、ハードウェアまたはソフトウェア、あるいはその組合せで実施す

ることができる。計算は、フレームごと（すなわちリアルタイム）、より低頻度で（ニア・リアルタイム）、またはオフラインで行うことができる。実施形態によっては、すべての計算をリアルタイムで行うか、または前述のように、特定のグラフィックス・データ・セットについていくつかの値を事前計算し、最終的な計算のみをリアルタイムで行うこともできる。他の実施形態では、すべての計算をオフラインで行うことができる。

【0124】

次に図19Bを参照すると、グラフィックス・データのセットの一実施形態が示されている。図に示すように、この例では、グラフィックス・データ・セット450は複数の汎用オブジェクト560A～Cを含む。各汎用オブジェクトはさらに複数のオブジェクト変形体570を含む。これらのオブジェクト変形体自体が複数のポリゴンおよび対応するレンダリング属性情報（たとえばテクスチャ）を含む。オブジェクト変形体は、異なる詳細レベル（LOD）に対応することができ、所定の最低フレーム・レートを実現するためにレンダリング・プロセスの前またはレンダリング・プロセス中（たとえば図19Aのステップ332）に選択することができる。たとえば、オブジェクト変形体576Aは、図1Bの球体に対応し、オブジェクト変形体576Bは図1Aの球体に対応する。さらに、オブジェクト変形体576Bは、球体にテクスチャ・マップする大理石テクスチャなどのレンダリング属性を含み、オブジェクト変形体576Bはテクスチャのない単純なシェーディングのレンダリング属性を含むことができる。オブジェクト変形体の中には、同じポリゴンを共有し、属性をレンダリングすることによってのみ変わるものもあるかもしれない。同様に、他のオブジェクト変形体も、同じレンダリング属性を共有する球体とすることができ、ポリゴン数により変わることができる。汎用オブジェクトによって、1つのオブジェクト変形体しかないものや、多数の変形体を持つものがある。

【0125】

レンダリング中、グラフィックス・システムは、レンダリングするすべての汎用オブジェクトの各オブジェクト変形体の推定レンダリング時間を計算し、次に、所定の最低フレーム・レートと整合する最も視覚的に現実感のあるオブジェク

ト変形体を選択するように構成することができる。特定の汎用オブジェクトのために選択されるオブジェクト変形体はフレームごとに変わることがあるが、グラフィックス・システムは、特定の汎用オブジェクトのためにオブジェクト変形体を選択する際にいくつかの要素を考慮に入れることができる。たとえば、グラフィックス・システムは、汎用オブジェクトの位置を考慮することができる。背景にある汎用オブジェクトには、より低い優先順位または重要度を与え、したがってよりレンダリングの速いオブジェクト変形体が選択されるようにすることができる。たとえばフライト・シミュレータでは、背景にある森の中の木に対応する汎用オブジェクトには、すぐ前景にある敵機に対応する別の汎用オブジェクトより低い優先順位を与える（それによって、グラフィックス・システムがよりレンダリングの速い、視覚的現実感がより少ないオブジェクト変形体を選択する可能性を高くする）ことができる。グラフィックス・システムは、オブジェクト変形体を選択する際にヒステリシスを使用するように構成することもできる。たとえば、直前の200フレームについて汎用オブジェクト560Cのためにオブジェクト変形体576Aを選択しており、次に、パフォーマンス限界によってグラフィックス・システムが所定の最低フレーム・レートを維持するためにオブジェクト変形体576Bを選択することを強制されたとする。1フレームのためにオブジェクト変形体576Bをレンダリングした後、システムのパフォーマンス限界では視覚的に現実感のあるオブジェクト変形体576Aを選択することができるようになった場合でも、次の数フレームのためにオブジェクト変形体576Bを選択し続けるようにグラフィックス・システムを構成することができる。このヒステリシスは、特定の汎用オブジェクトのために異なるオブジェクト変形体を急速に連続して選択した場合に発生することがある望ましくないフリッカを防止することができるので有利である。

【0126】

射影の不変性の実証

単位面積ワールド空間ポリゴンをランダムな視点から垂直に見ると仮定すると、射影面積を計算するのに、重要なのはポリゴンのファセット垂直線と目視方向との間の角度のみである。所与のビューのための射影ポリゴンの画面空間面積は

、この角度の余弦（正規化ベクトルの内積）に過ぎない。したがって、可能なすべてのビューの均等な確率分布は、方向の一樣分布に過ぎない。これは、単位球面上に一樣に分布した点として表すことができる。目視方向の半分は背面を向きであるため、普遍性を失うことなく、点から成る半球を考慮に入れるだけで十分である。

【0127】

同じ内積を有し、同じ面積を生じさせるビューに対応する点は、すべてこの半球上の同じ緯度にある。より低い緯度の「厚さ」によって、円周の変化がまったく相殺され、その結果視角（したがって面積）の均一な確率が得られる。

【0128】

図20に、この単位半球の断面図を示す。緯度 θ における一定した高さの帯域の測定値 da は $2\pi \cdot \sin \theta \cdot d\theta$ である。しかし、 $d\theta = da / (\sin \theta)$ であり、したがってこれらの正弦は相殺され、いずれの帯域内の点も θ とは独立する。しかし、この不変性の結果は三次元においてのみ成り立つことに留意されたい。

【0129】

コンピュータ・ネットワークの例—図21

図21に、少なくとも1つのサーバ・コンピュータ502と1つまたは複数のクライアント・コンピュータ506A~N（図21に示す実施形態ではクライアント・コンピュータ506A~Bが図示されている）を含む例示のコンピュータ・ネットワーク500を示す。クライアント・システムのうちの1つまたは複数のクライアント・システムを、前述のように各コンピュータ・システムが1つまたは複数の前述のグラフィックス・システム112を有するコンピュータ・システム80と同様に構成することができる。各クライアント・システムは、さらに上述のように時間推定のレンダリングを実行するように構成することもできる。サーバ502とクライアント506は、ローカル・エリア・ネットワーク（LAN）、ワイド・エリア・ネットワーク（WAN）、インターネット接続などの様々な接続機構504を介して結合することができる。一実施形態では、サーバ502は3-D幾何形状データ（圧縮も可）を記憶し、1つまたは複数のクライア

ント506に送ることができる。クライアント506は、3-D幾何形状データを受け取り、(必要な場合は)圧縮解除し、レンダリング時間を推定し、次にその幾何形状データを(必要に応じて修正されたレンダリング・パラメータを使用して)レンダリングする。本明細書で使用するレンダリング・パラメータは、レンダリングするオブジェクト/シーン/画像内のピクセル数またはサンプル数、1ピクセル当たりのサンプル数、カラー階調、テクスチャ・パラメータ、ライト数(およびそれらに対応する特性)、特殊レンダリング効果(たとえば透明、アンチエイリアシング、フォギング、ぼかし効果)およびレンダリングするオブジェクト数を含むことに留意されたい。レンダリングされた画像は次にクライアントの表示装置上に表示される。クライアントは、前述のように標準またはスーパーサンプル型サンプル・バッファを使用して、幾何形状データをレンダリングし、画像を表示することができる。他の実施形態では、圧縮された3-D幾何形状データをクライアントコンピュータ506間で転送することができる。

【0130】

上述の実施形態についてかなり詳細に説明したが、他の実施形態も可能である。当業者なら上述の説明を完全に理解すれば、多くの変形態様および変更が明らかになるであろう。特許請求の範囲は、このような変形態様および変更をすべて含むものと解釈されることを意図している。本明細書で使用した見出しは、構成のためのものに過ぎず、本明細書に記載の説明または特許請求の範囲を限定することを意味するものではない。

【0131】

産業上の利用

上述の方法およびコンピュータ・ソフトウェア・プログラムは、コンピュータ・ハードウェア、グラフィックス装置、パーソナル・ディジタル・アシスタント(PDA)、グラフィックス表示装置を備えた機器、グラフィックス・プロセッサ、グラフィックス・サブシステム、仮想現実システム、ヘッドマウント表示装置、コンピュータ・ゲーム、および3Dグラフィックス・データの生成、表示、送信または操作を行うその他の電子装置で利用することができる。この方法およびコンピュータ・ソフトウェア・プログラムは、ハードウェア、ソフトウェア、ま

たはそれらの任意の組合せで実施することができる。

【図面の簡単な説明】

【図1A】

少数のポリゴンを使用してモザイク化された球体を示す図である。

【図1B】

より多数のポリゴンを使用してモザイク化された球体を示す図である。

【図2】

グラフィックス・システムの考えられる1つのパフォーマンス限界曲線を示すグラフである。

【図3】

コンピュータ・システムの例を示す図である。

【図4】

図3のコンピュータ・システムの略ブロック図である。

【図5】

図4のグラフィックス・システムの一実施形態のをより詳細に示すブロック図である。

【図6】

5個の異なる三次元レンダリングされたオブジェクトの画像である。

【図7】～【図11】

図6に示す各オブジェクトの三角形パラメータ化を示すグラフである。

【図12】

三角形のアスペクト比および歪みの計算を示す図である。

【図13】～【図14】

図6のうちの2つのオブジェクトのモデル空間および画面空間内のアスペクト比のヒストグラムを示すグラフである。

【図15】

モデル空間から画面空間への変換を示す図である。

【図16】

図6からのTレックスについての計算を示す図である。

【図 17】

人間の眼の領域を示す図である。

【図 18】

様々な表示装置の特性評価を示す図である。

【図 19A】

パフォーマンス限界を推定する方法の一実施形態を示すフローチャートである。

【図 19B】

グラフィックス・データ・セットの一実施形態を示す図である。

【図 20】

モデル空間から画面空間への射影の不変性の証明を示す図である。

【図 21】

複数のコンピュータを接続するコンピュータ・ネットワークの一実施形態を示す図である。

【図 1A】

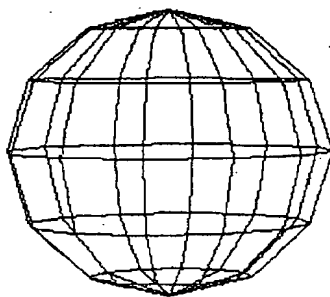


FIG. 1A

【図 1 B】

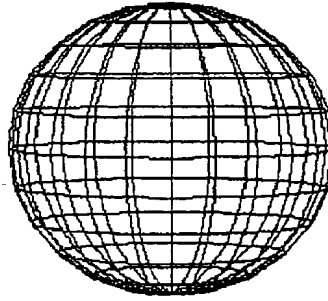
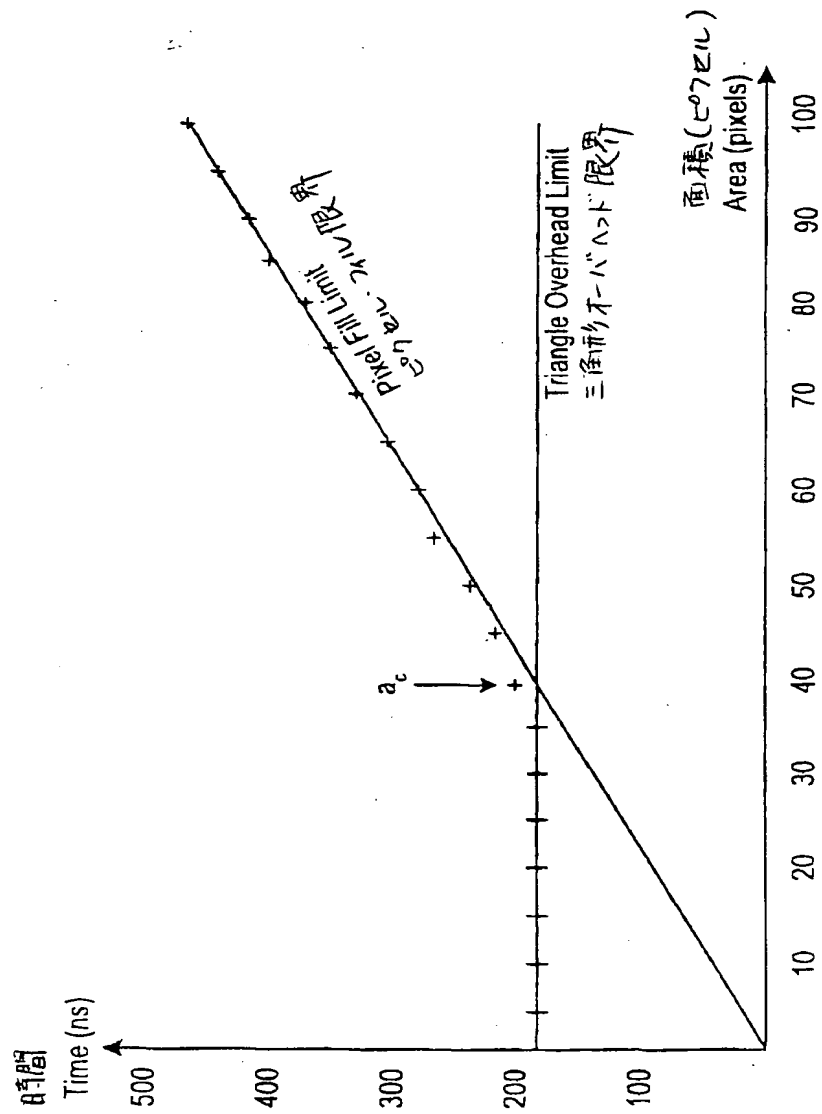


FIG. 1B

【図 2】



【図3】

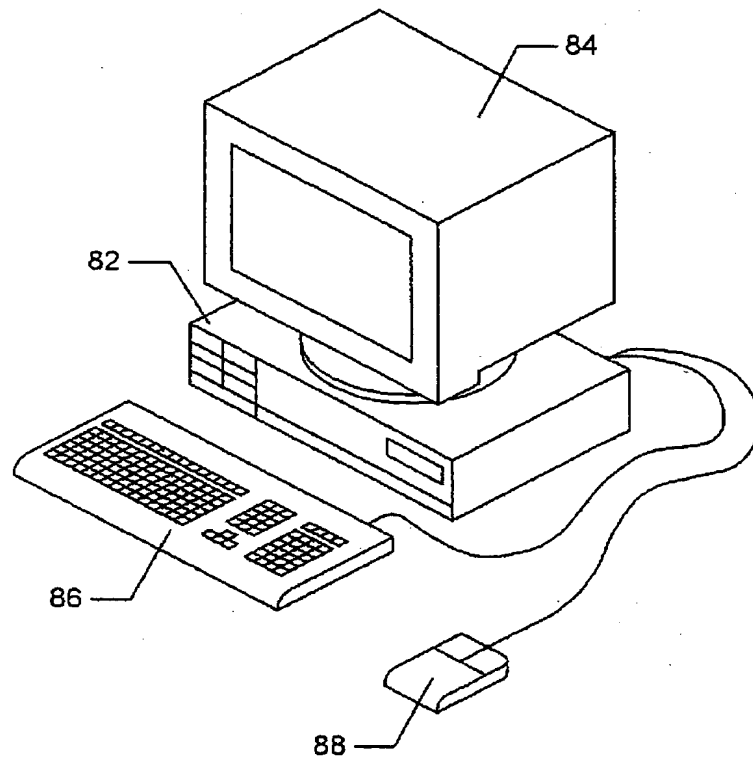
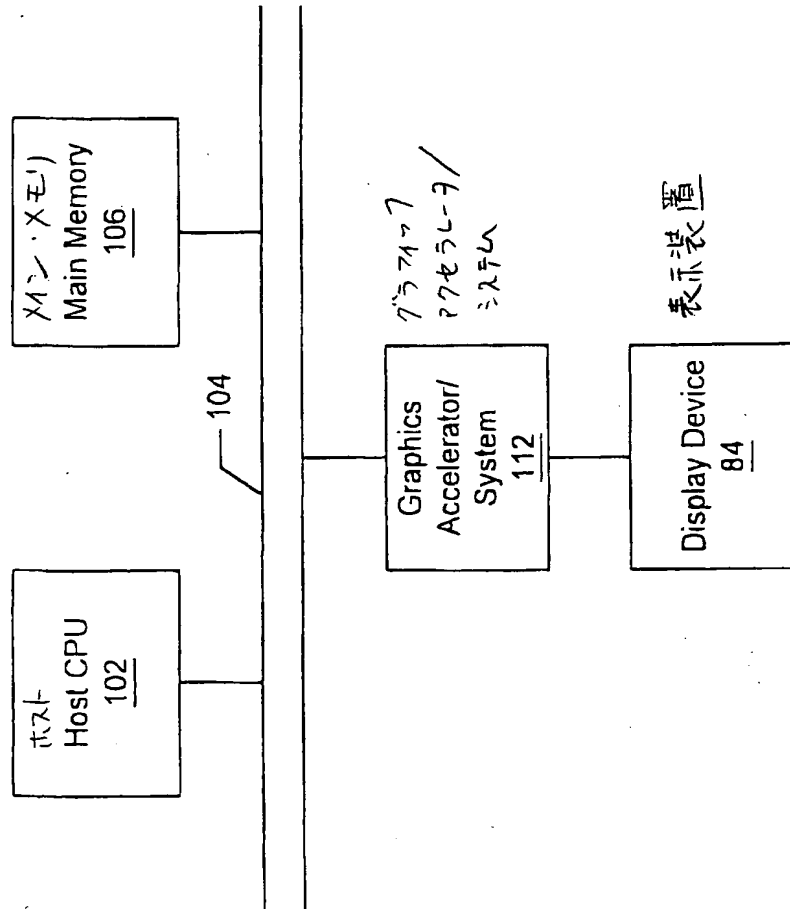
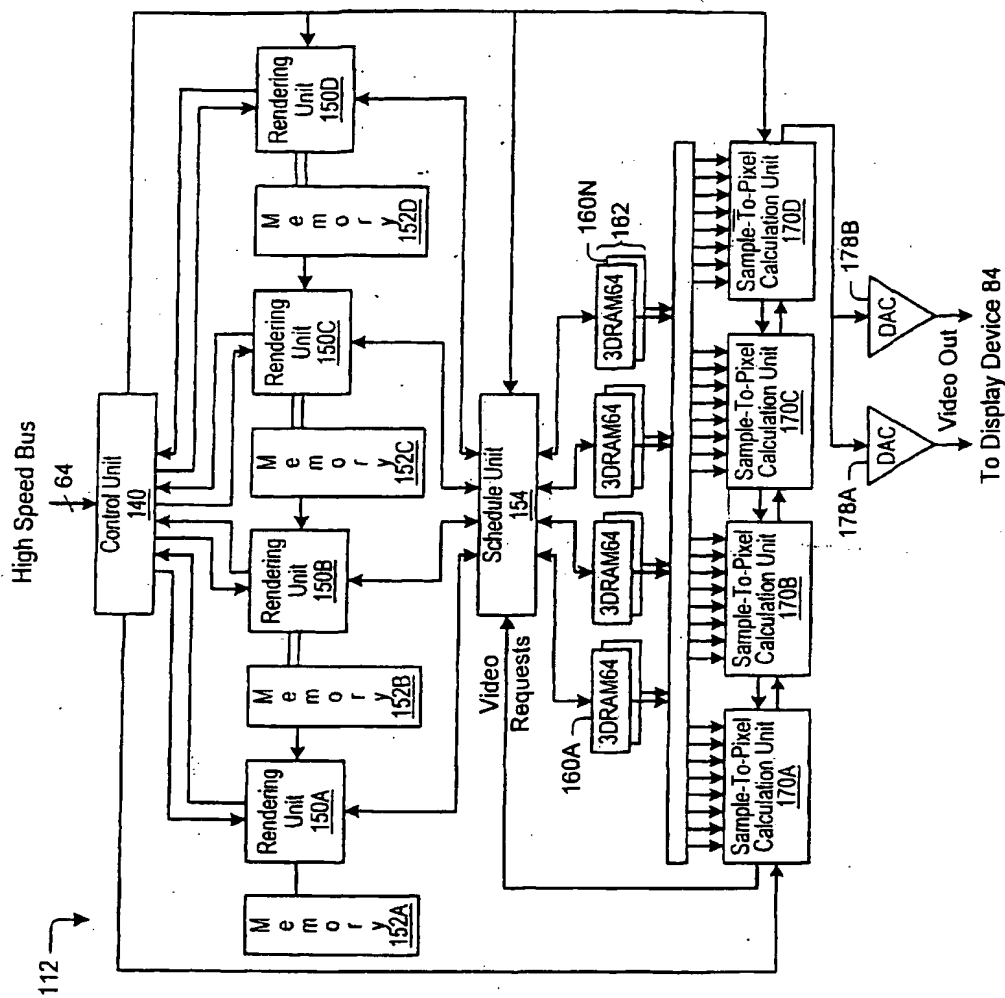


FIG. 3

【図4】



【図 5】



140 制御ユニット

152A メモリ

150A レンダリング・ユニット

152B メモリ

150B レンダリング・ユニット

152C メモリ

150C レンダリング・ユニット

152D メモリ

150D レンダリング・ユニット

154 スケジュール・ユニット

170A サンプル-ピクセル計算ユニット

170B サンプル-ピクセル計算ユニット

170C サンプル-ピクセル計算ユニット

170D サンプル-ピクセル計算ユニット

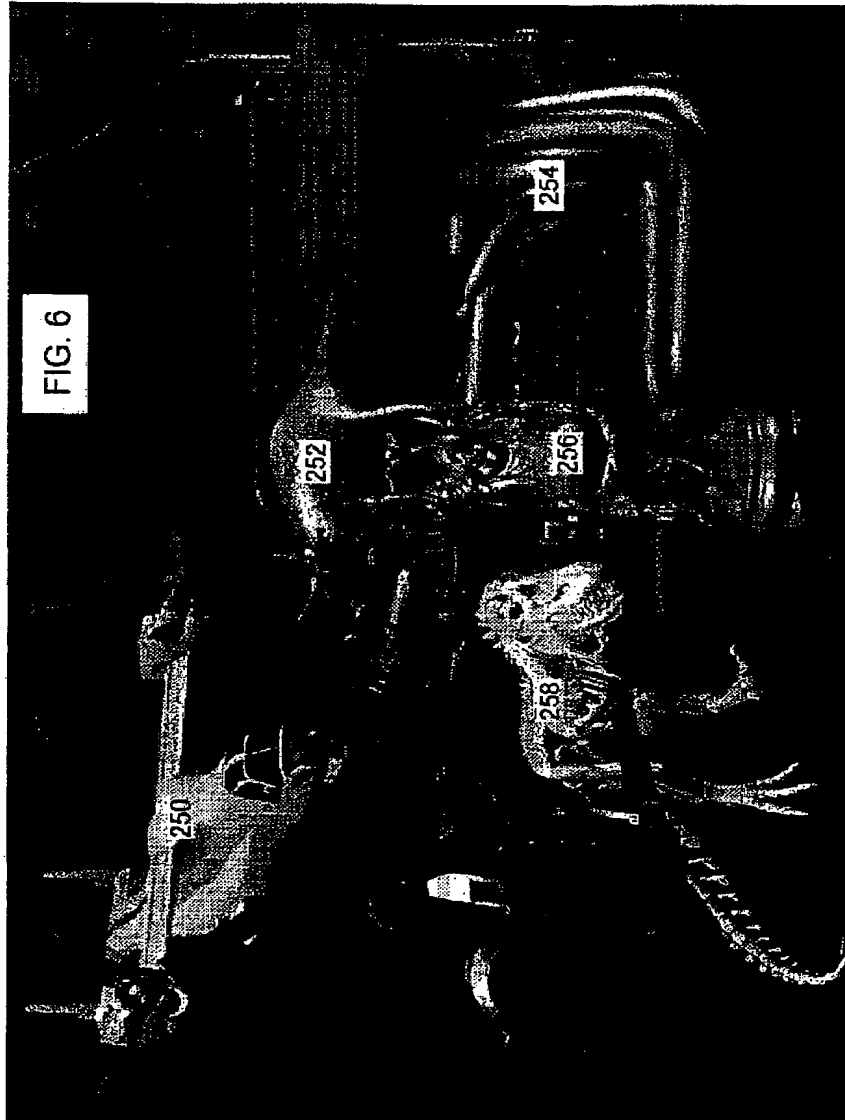
High Speed Bus 高速バス

Video Request ビデオ要求

Video Output ビデオ出力

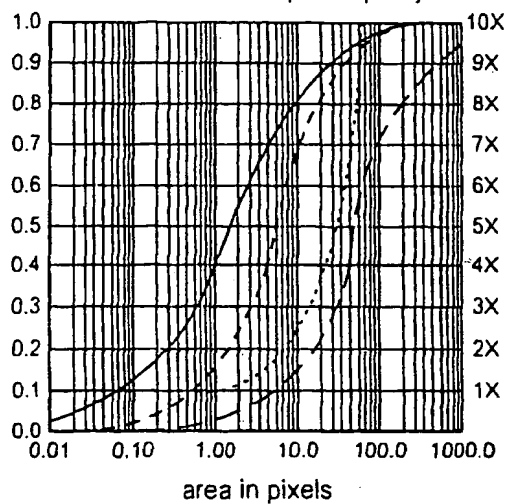
To Display Device 表示装置 84 へ

【図6】



【図 7】

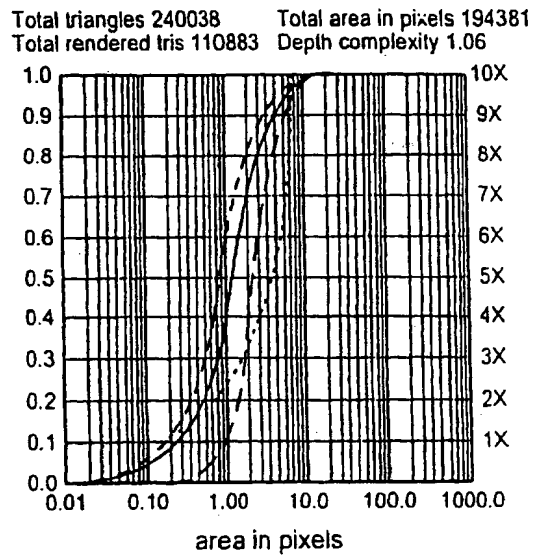
Total triangles 30868 Total area in pixels 203848
 Total rendered tris 14790 Depth complexity 1.47



Screen space triangle area
 statistics for Osprey

Total triangles	三角形総数
Total area in pixels	ピクセル数での総面積
Total rendered tris	レンダリング済三角形総数
Depth complexity	デプス複雑度
area in pixels	ピクセル数での面積
Screen space triangle	オスプリーの画面空間
statistics for Osprey	三角形面積統計

【図 8】

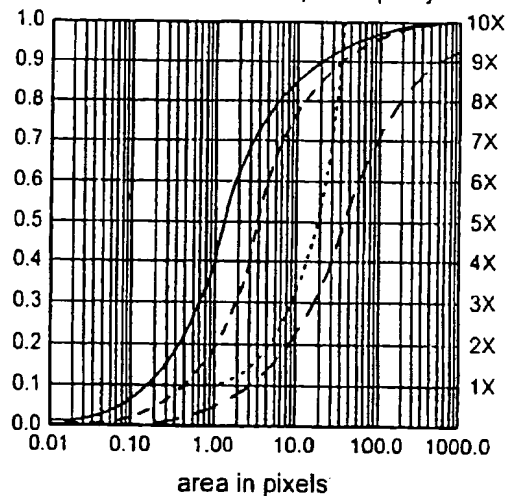


Screen space triangle area
 statistics for Textured Triceratops

Total triangles	三角形総数
Total area in pixels	ピクセル数での総面積
Total rendered tris	レンダリング済三角形総数
Depth complexity	デプス複雑度
area in pixels	ピクセル数での面積
Screen space triangle statistics for Textu- red Triceratops	テクスチャ化トリケ のラトプスの画面空 間三角形面積統計

【図 9】

Total triangles 254478 Total area in pixels 1004538
 Total rendered tris 124596 Depth complexity 4.79

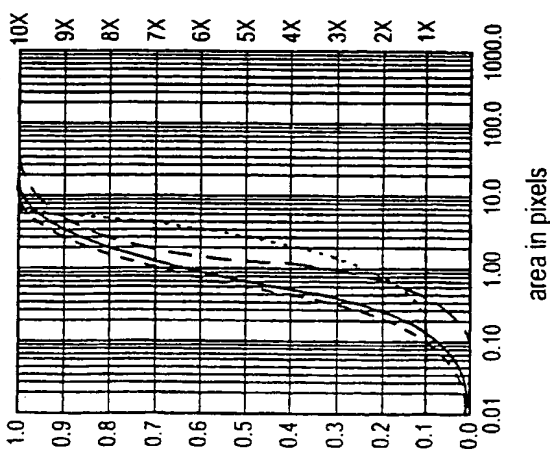


Screen space triangle area
 statistics for Engine

Total triangles	三角形総数
Total area in pixels	ピクセル数での総面積
Total rendered tris	レンダリング済三角形総数
Depth complexity	デプス複雑度
area in pixels	ピクセル数での面積
Screen space triangle statistics for Engine	エンジンの画面空間 三角形面積統計

【図 10】

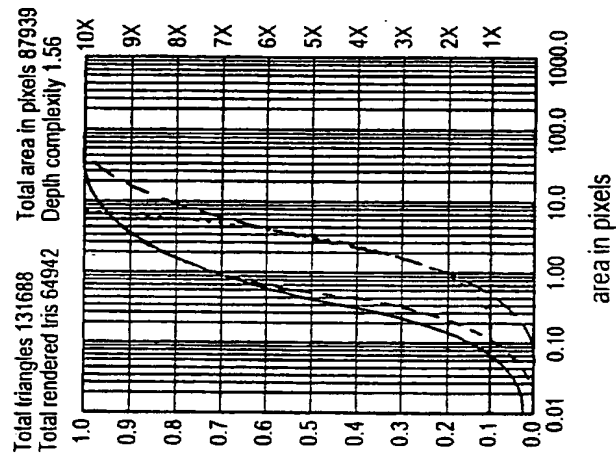
Total triangles 293232 Total area in pixels 150103
 Total rendered tris 141419 Depth complexity 1.18



Screen space triangle area
 statistics for Buddha

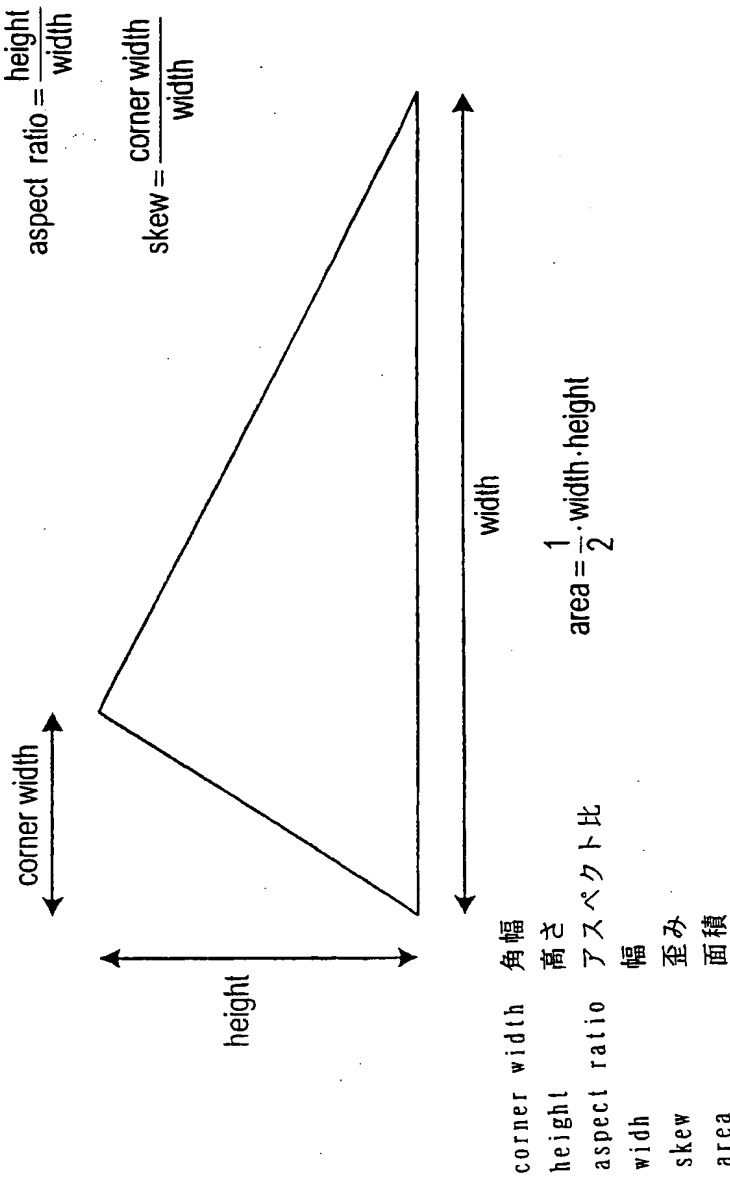
Total triangles	三角形総数
Total area in pixels	ピクセル数での総面積
Total rendered tris	レンダリング済三角形総数
Depth complexity	デプス複雑度
area in pixels	ピクセル数での面積
Screen space triangle statistics for Buddha	仏像の画面空間 三角形面積統計

【図 1 1】

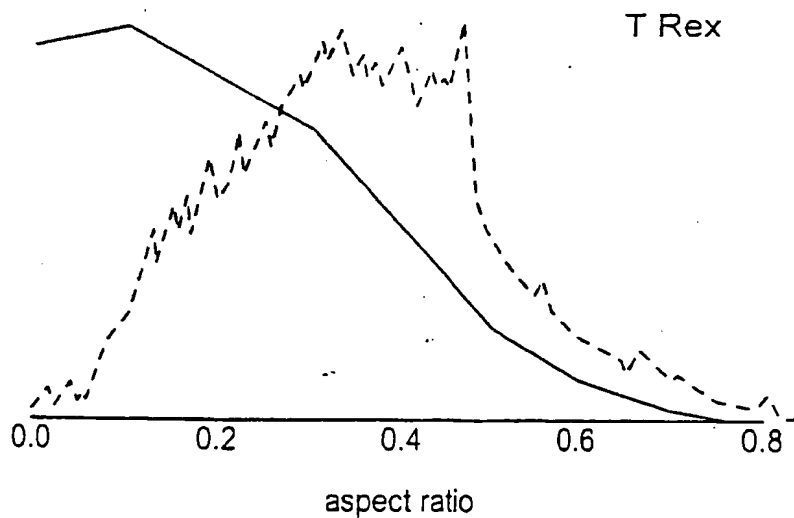

 Screen space triangle area
 statistics for T Rex

Total triangles	三角形総数
Total area in pixels	ピクセル数での総面積
Total rendered tris	レンダリング済三角形総数
Depth complexity	デプス複雑度
area in pixels	ピクセル数での面積
Screen space triangle statistics for T Rex	T レックスの画面空間 三角形面積統計

【図 12】



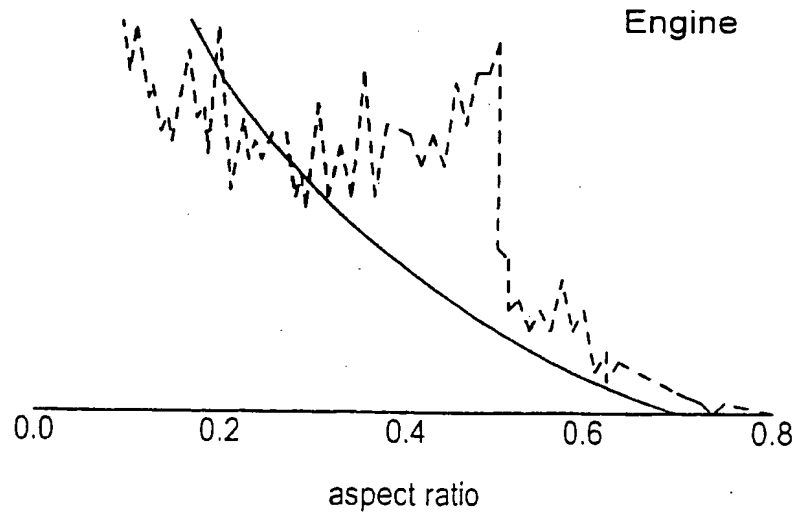
【図 13】



Triangle aspect ratio statistics for T Rex.

T Rex	T レックス
aspect ratio	アスペクト比
Triangle aspect ratio	T レックスの三角形
statistics for T Rex	アスペクト比統計

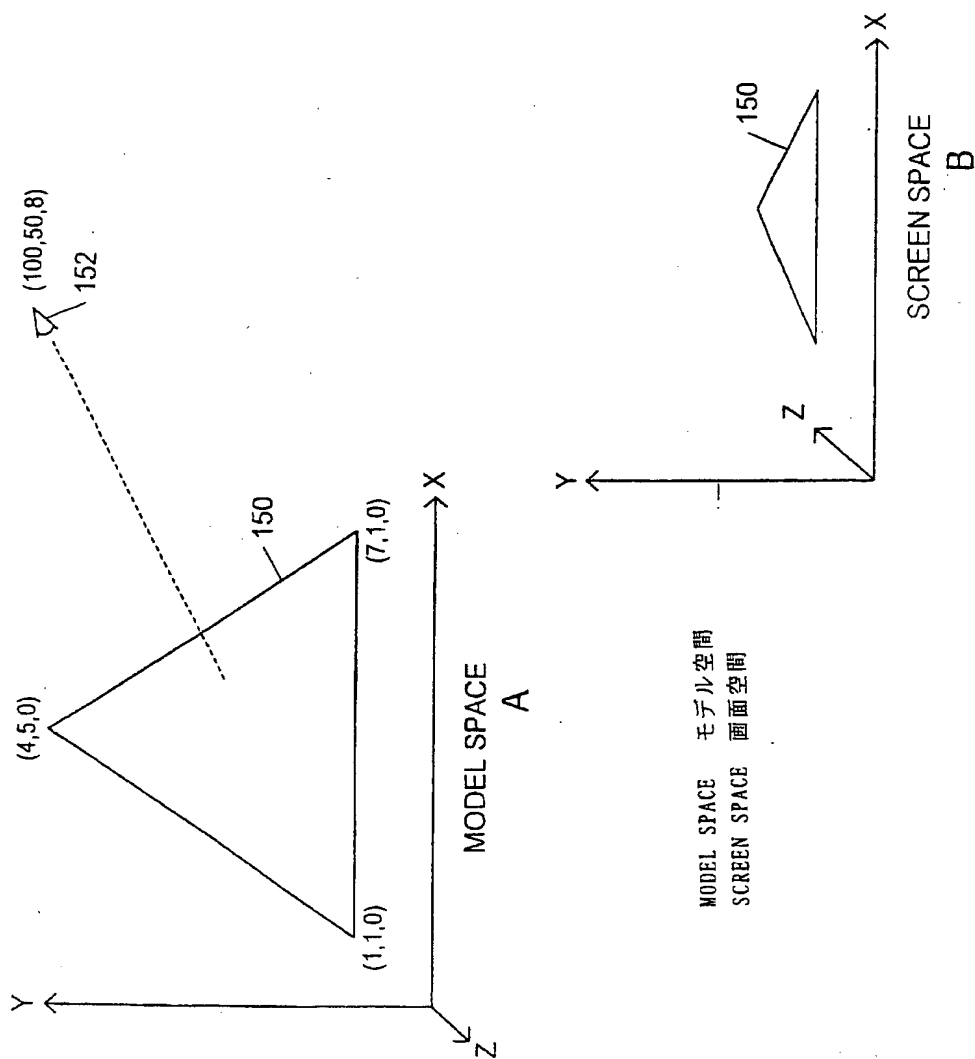
【図 14】



Triangle aspect ratio statistics for Engine.

Engine	エンジン
aspect ratio	アスペクト比
Triangle aspect ratio statistics for Engine	エンジンの三角形アスペクト比統計

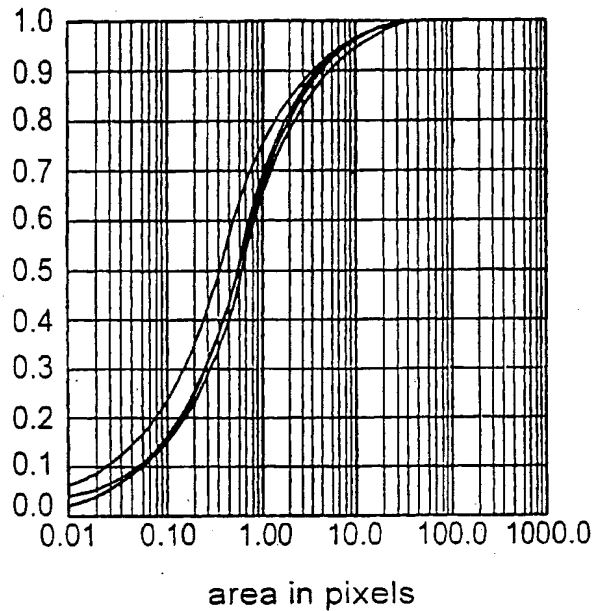
【図 15】



【図16】

T Rex

Total triangles 131688
Total rendered tris 65844



Screen space prediction of $f(a)$
based on model space.

T Rex

Total triangles

Total area in pixels

area in pixels

Screen space prediction of
 $f(a)$ based on model space

Tレックス

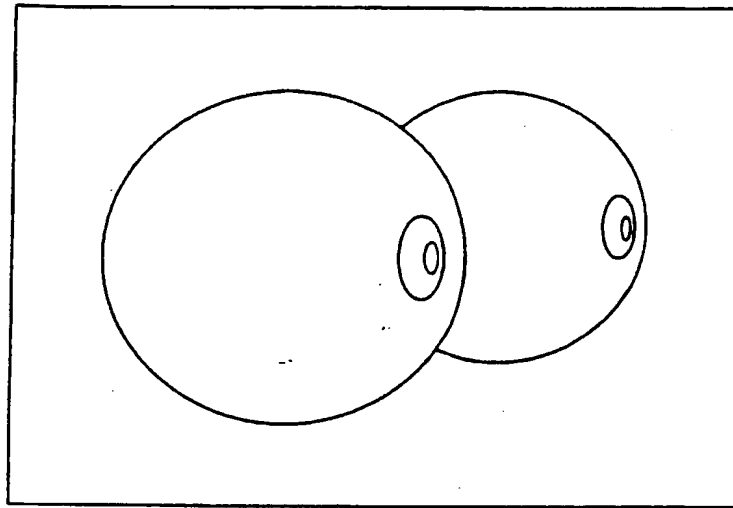
三角形総数

ピクセル数での総面積

ピクセル数での面積

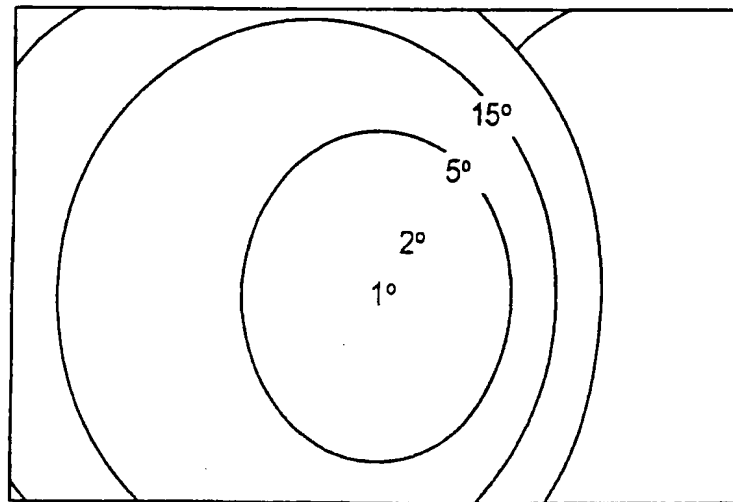
モデル空間に基づく
 $f(a)$ の画面空間予測

【図 17】



Human eye variable density regions.

A



Close up of Human eye variable density regions

B

Human eye variable density regions

人間の目の可変密度領域

Close up of Human eye
variable density regions人間の目の可変密度領域
のクローズアップ

【図 18】

Display Device 400	Display Pixel Resolution 402	Display pixel sz in min 404	Display FOV in steradians 406	Pixels: 0.47 min limit 408	Pixels: 1.5 min limit 410	Pixels: eye limit 412	Pixels: display limit 414	Pixels: eye & display limit 416	Triangles: eye & display limit 418
18" CRT at 24"	1280x1024	1.6	0.26	14.1M	1.37M	1.59M	1.31M	1.06M	0.38B
24" CRT at 24"	1920x1200	1.4	0.36	19.4M	1.88M	1.92M	2.30M	1.55M	0.56B
37" CRT at 24"	1280x1024	3.0	0.73	39.9M	3.86M	3.20M	1.31M	0.94M	0.34B
65 °HMD*	1280x1024	2.9	0.73	39.4M	3.81M	3.17M	1.31M	1.01M	0.73B
95 °HMD*	1280x1024	4.0	1.17	63.6M	6.16M	4.68M	1.31M	0.88M	0.63B
31" TV at 11'	640x480	1.0	0.03	1.4M	0.14M	0.40M	0.31M	0.28M	0.1B
60" HDTV @ 8'	1920x1080	1.0	0.16	8.7M	0.84M	1.17M	2.07M	1.07M	0.38B
50" Movie @ 65'	2350x1000	1.0	0.20	10.8M	1.04M	1.30M	2.35M	1.16M	0.42B
Human eye FOV*	na	na	4.16	226.0M	21.9M	14.78M	na	na	10.64B
Full Sphere FOV*	na	na	12.57	682.0M	66.0M	na	na	na	491.00B
*stereo.	M. All pixel numbers are in units of millions of pixels. B. Triangles are in units of billions of triangles per second.								

400 表示装置 402 表示ピクセル分解能 404 表示ピクセル・サイズ (分)

406 表示 FOV (ステラジアン) 408 ピクセル : 0.47 分限界

410 ピクセル : 1.5 分限界 412 ピクセル : 目の限界

414 ピクセル : 表示装置の限界 416 ピクセル : 目の限界

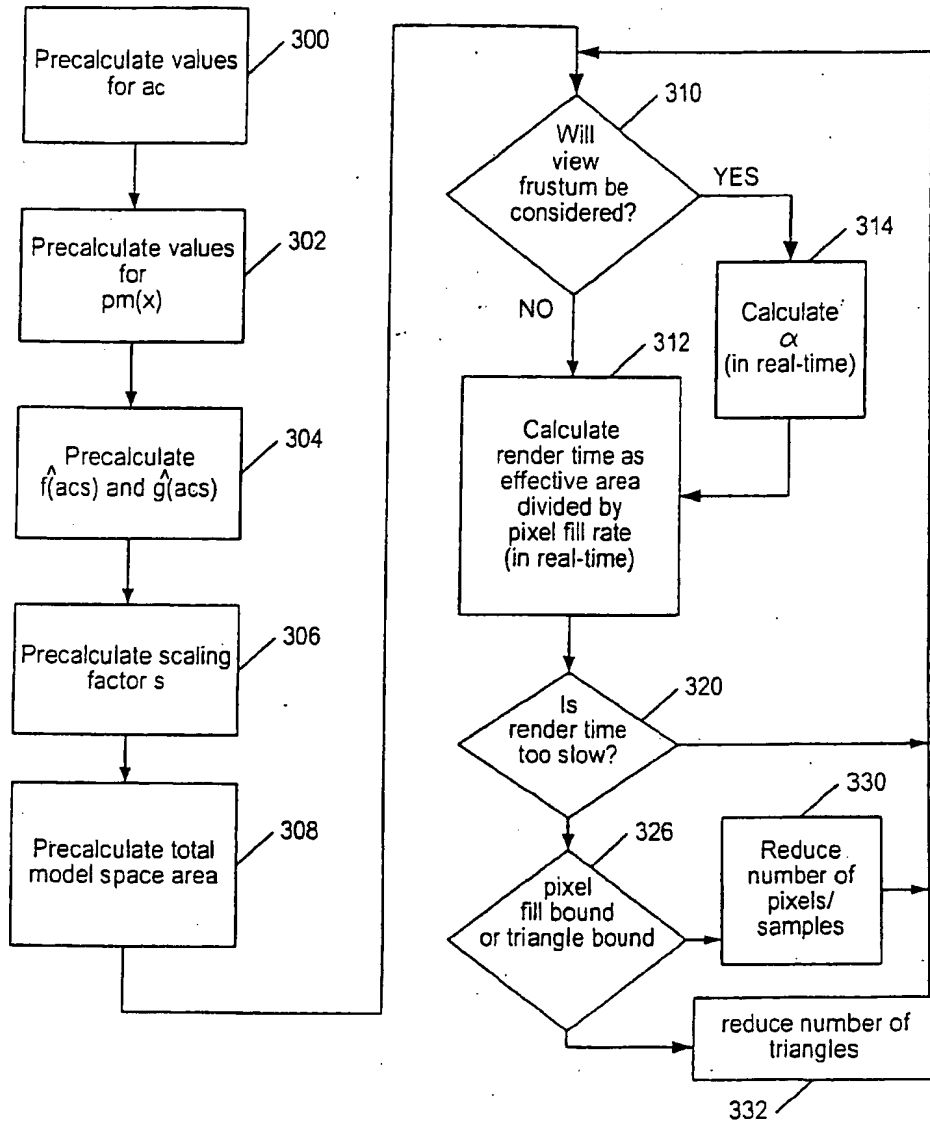
418 ピクセル : 表示装置の限界

M. 三角数はすべて 100 万ピクセル単位

B. 三角数はすべて 10 億個単位

Movie 映画、Human eye 人間の目、Full Sphere 全球面の FOV、stereo 立体

【図 19 A】



3 0 0 a.c. の値を事前計算する
 3 0 2 $pm(x)$ の値を事前計算する
 3 0 4 $f(acs)$ および $g(acs)$ の値を事前計算する
 3 0 6 倍率 s を事前計算する
 3 0 8 倍率 s を事前計算する
 3 1 0 倍率 s を事前計算する
 3 1 4 倍率 s を事前計算する
 3 1 2 倍率 s を事前計算する
 3 2 0 倍率 s を事前計算する
 3 2 6 倍率 s を事前計算する
 3 3 0 倍率 s を事前計算する
 3 3 2 倍率 s を事前計算する

【図 20】

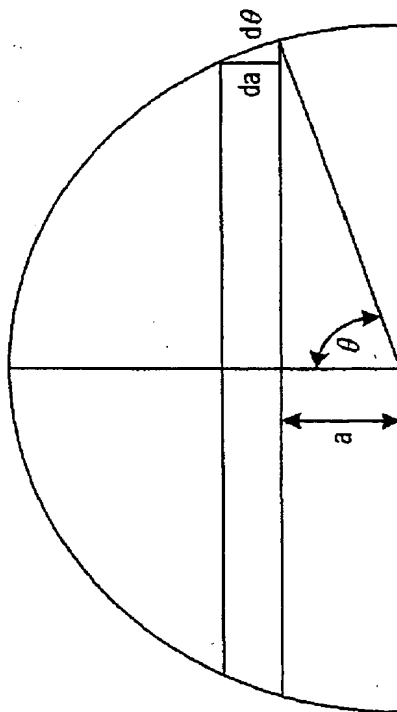


FIG. 20

【図 21】

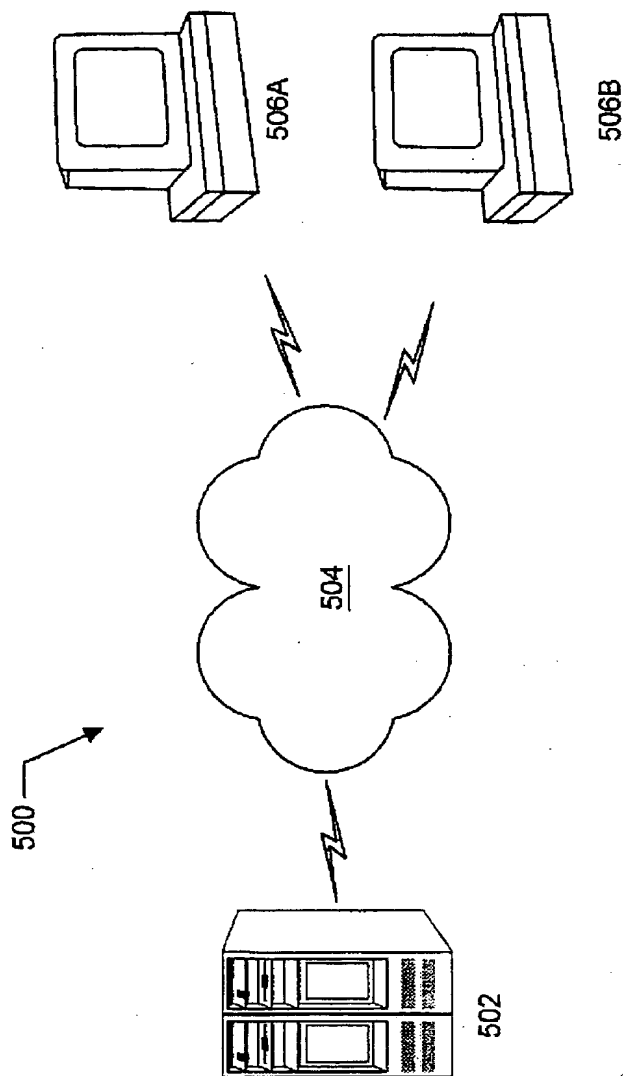


FIG. 21

【手続補正書】特許協力条約第34条補正の翻訳文提出書

【提出日】平成12年2月22日(2000. 2. 22)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 コンピュータ・ソフトウェア・プログラムがポリゴン・レンダリングのためのグラフィックス・システムのパフォーマンスを推定するように構成され、搬送媒体に記録された、グラフィックス・システムにおけるレンダリング時間を推定するコンピュータ・ソフトウェア・プログラムであって、

グラフィックス・データが異なるレンダリング属性を有する複数のオブジェクト変形体を有する、1つまたは複数の汎用オブジェクトを含むグラフィックス・データのセットのレンダリング時間を計算し、前記レンダリング時間が所定の最低フレーム・レートによって指定されたレンダリング時間を超える場合、

前記グラフィックス・データがポリゴン・オーバーヘッド制限付きであるかピクセル・フィル制限付きであるかを判断し、

前記グラフィックス・データがピクセル・フィル制限付きであるのに応答してより少ない合計ピクセル計算を必要とするオブジェクト変形体を選択し、前記グラフィックス・データがポリゴン・オーバーヘッド制限付きであるのに応答してより少ないポリゴンを有するオブジェクト変形体を選択することによって、前記所定の最低フレーム・レートを達成するようにレンダリング・パラメータを変更するように構成された複数の命令を含む、コンピュータ・ソフトウェア・プログラム。

【請求項2】 各オブジェクト変形体が複数のポリゴンとレンダリング属性とを含み、前記複数の命令が、レンダリングする各汎用オブジェクトのために1つの変形を選択してレンダリングするようにさらに構成された請求項1に記載のコンピュータ・ソフトウェア・プログラム。

【請求項3】 前記複数の命令が、各汎用オブジェクトの各変形体のレンダリング時間を計算し、レンダリングする各汎用オブジェクトのために前記所定の最低フレーム・レートに整合した視覚的に最も現実感のあるオブジェクトを選択するようにさらに構成された請求項2に記載のコンピュータ・ソフトウェア・プログラム。

【請求項4】 前記複数の命令が、前記所定の最低フレーム・レートとの整合性、ヒステリシス、各オブジェクトの相対的重要度、および各オブジェクトの位置の考慮項目のうちの1つまたは複数の考慮項目に基づいて、レンダリングする各汎用オブジェクトのために視覚的に最も現実感のあるオブジェクト変形体を選択するようにさらに構成された請求項2または3に記載のコンピュータ・ソフトウェア・プログラム。

【請求項5】 汎用オブジェクトのサブセットに他の汎用オブジェクトより高い優先順位が与えられる請求項1に記載のコンピュータ・ソフトウェア・プログラム。

【請求項6】 前記複数の命令が、前記オブジェクト変形体のうちの少なくとも1つのオブジェクト変形体の累積確率密度分布 f_a を計算するようにさらに構成され、 f_a が a 以下の面積を有する前記オブジェクト変形体内のランダムに選定されたポリゴンの確率である請求項1、2、3、または5に記載のコンピュータ・ソフトウェア・プログラム。

【請求項7】 前記複数の命令が、前記オブジェクトの変形体のうちの少なくとも1つのオブジェクト変形体内のすべてのポリゴンの累積面積 g_a を計算するようにさらに構成され、 g_a が前記オブジェクト変形体の合計表面積に対する、面積 a を有する前記オブジェクト変形体内のポリゴンによって占められる合計表面積の量の割合である請求項6に記載のコンピュータ・ソフトウェア・プログラム。

【請求項8】 前記複数の命令が、複数の異なる視角にわたって f_a を平均するようにさらに構成された請求項6に記載のコンピュータ・ソフトウェア・プログラム。

【請求項9】 前記複数の命令が、

前記汎用オブジェクトのうちの1つまたは複数の汎用オブジェクトのための可能な視点を複数の異なる視角セットに分け、

異なる各視角セットのための f_a の平均値を計算し、

前記レンダリング時間の計算に使用するために現行視点に基づいて f_a の前記平均値のうちの1つの値を選択するようにさらに構成された請求項6に記載のコンピュータ・ソフトウェア・プログラム。

【請求項10】 前記複数の命令が、複数の異なる視角にわたって g_a を平均するようにさらに構成された請求項7、8、または9に記載のコンピュータ・ソフトウェア・プログラム。

【請求項11】 前記複数の命令が、各ポリゴンのアスペクト比を計算するようにさらに構成され、各ポリゴンの前記アスペクト比が前記ポリゴンの高さを前記ポリゴンの幅で割った割合である請求項7に記載のコンピュータ・ソフトウェア・プログラム。

【請求項12】 前記複数の命令が、各ポリゴンの歪みを計算するようにさらに構成され、各ポリゴンの前記歪みが前記ポリゴンの角幅を前記ポリゴンの幅で割った値である請求項11に記載のコンピュータ・ソフトウェア・プログラム。

【請求項13】 f_a 、 g_a 、アスペクト比、および歪みが、モデル空間においてオフラインで1回と画面空間のためにリアルタイムで1回の、それぞれ2回計算される請求項12に記載のコンピュータ・ソフトウェア・プログラム。

【請求項14】 前記複数の命令が、各オブジェクト変形体の複数の異なる視点についてレンダリング時間を計算し、次に、各オブジェクト変形体の前記複数の異なる視点について計算されたレンダリング時間を平均するようにさらに構成された請求項1、2、12、または13に記載のコンピュータ・ソフトウェア・プログラム。

【請求項15】 前記複数の命令が、各オブジェクト変形体の可能なすべての視点についてレンダリング時間を計算し、次に、計算されたレンダリング時間を平均するように構成された請求項14に記載のコンピュータ・ソフトウェア・プログラム。

【請求項16】 前記複数の命令が、前記オブジェクト変形体の対称な部分の代表的な視点のレンダリング時間を計算してから、計算されたレンダリング時間を平均することによって、前記オブジェクト変形体の可能なすべての視点についてレンダリング時間を計算するように構成された請求項15に記載のコンピュータ・ソフトウェア・プログラム。

【請求項17】 前記複数の命令が、畳込み計算を行うことによって前記オブジェクト変形体の可能なすべての視点についてレンダリング時間を計算し、平均するようにさらに構成された請求項15または16に記載のコンピュータ・ソフトウェア・プログラム。

【請求項18】 前記複数の命令が、前記所定の最低フレーム・レート達成するために、サンプル密度、1ピクセル当たりのサンプル数、ピクセル数、照明効果、光源数、詳細レベル、ポリゴン数、アンチエイリアシング、フォギング、テクスチャ・マッピング・パラメータ、プログラム可能シェーダ、シェーディング・パラメータ、およびカラー階調のレンダリング・パラメータのうちの1つまたは複数のレンダリング・パラメータを変更するようにさらに構成された請求項1、2、または16に記載のコンピュータ・ソフトウェア・プログラム。

【請求項19】 前記搬送媒体がコンピュータ可読媒体または伝送媒体である請求項1または2に記載のコンピュータ・ソフトウェア・プログラム。

【請求項20】 幾何形状データの特定のセットのためのグラフィックス・システムのレンダリング・パフォーマンスを推定する方法であって、

前記グラフィックス・システムのピクセル・フィル・レートを判断するステップと、

前記幾何形状データが異なる数のポリゴンを含む複数のオブジェクト変形体を有する複数の汎用オブジェクトを含み、1フレーム当たりのレンダリング時間が前記幾何形状データの有効面積を前記ピクセル・フィルレートで割った値であり、前記有効面積が前記幾何形状内のすべての正面向きポリゴンの面積の和に等しく、所定の面積 a_c 未満の面積を有する前記幾何形状内の各正面向きポリゴンの面積が前記所定の面積 a_c に切り上げられ、前記所定の面積 a_c が前記グラフィックス・システムのパフォーマンスを記述する定数である、前記グラフィックス

・システムの前記幾何形状データのための1フレーム当たりのレンダリング時間を計算するステップと、

前記1フレーム当たりのレンダリング・レートが所定の最低値を下回り、前記グラフィックス・システム・パフォーマンスがポリゴン・オーバーヘッド制限付きである場合、レンダリングのためにより少ないポリゴン数を有するオブジェクト変形体を選択するステップとを含む方法。

【請求項21】 前記1フレーム当たりのレンダリング時間が背面向きポリゴンのための調整値を含み、前記調整値が、背面向きポリゴンの数に所定の面積 a_0 を乗じた近似値および背面処理時間と最小ポリゴン・レンダリング時間との比である請求項20に記載の方法。

【請求項22】 前記1フレーム当たりのレンダリング・レートが所定の最低値を下回り、前記グラフィックス・システム・パフォーマンスがピクセル・フィル制限付きである場合、1ピクセル当たりの計算の数を減らすレンダリング属性を有するオブジェクト変形体を選択する請求項20または21に記載の方法。

【請求項23】 前記1フレーム当たりのレンダリング時間の結果として所定の最低値を下回るフレーム・レートになり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きである場合、前記グラフィックス・システムにグラフィックス・データの前記セットのピクセル面積を小さくさせるステップをさらに含む請求項21に記載の方法。

【請求項24】 前記グラフィックス・システムがサンプル密度を変えることができ、前記方法が、前記1フレーム当たりのレンダリング時間の結果として所定の最低値を下回るフレーム・レートになり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きである場合、前記グラフィックス・データの少なくとも一部をレンダリングするために使用される前記グラフィックス・システムのサンプル密度を低下させるステップをさらに含む請求項21に記載の方法。

【請求項25】 前記1フレーム当たりのレンダリング・レートが所定の最低値より下がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きである場合、前記グラフィックス・データの少なくとも一部に対し

て行われるテクスチャ計算の複雑度を低下させることによって前記グラフィックス・データのピクセル面積を減らすステップをさらに含む請求項21または24に記載の方法。

【請求項26】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがポリゴン・オーバーヘッド制限付きでない場合、レンダリングのためのより多いポリゴン数を有するセット・グラフィックス・データを選択するステップをさらに含む請求項21に記載の方法。

【請求項27】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システムがフィル・レート制限付きでない場合、前記グラフィックス・システムに前記グラフィックス・データのピクセル面積を増やさせるステップをさらに含む請求項21または26に記載の方法。

【請求項28】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きでない場合、ビデオ・サイズ変更によって前記グラフィックス・データのピクセル面積を増やすステップをさらに含む請求項21または26に記載の方法。

【請求項29】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きでない場合、1つまたは複数の背景オブジェクトを削除することによって前記グラフィックス・データのピクセル面積を減らすステップをさらに含む請求項21または26に記載の方法。

【請求項30】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがフィル・レート制限付きでない場合、前記グラフィックス・データの少なくとも一部をレンダリングするために使用されるサンプル密度を低下させるステップをさらに含む請求項21または26に記載の方法。

【請求項31】 前記1フレーム当たりのレンダリング・レートが所定の最大値より上がり、前記グラフィックス・システム・パフォーマンスがフィル・レ

ート制限付きでない場合、前記グラフィックス・データの少なくとも一部に対して行われるテクスチャ計算の複雑度を低下させることによって前記グラフィックス・データの画素面積を減らすステップをさらに含む請求項21または26に記載の方法。

【請求項32】 グラフィックス・システムのピクセル・フィル・レートを判断するステップと、

有効面積が前記グラフィックス・システムのピクセル・フィル限界およびポリゴン・オーバーヘッド限界を考慮したグラフィックス・データのレンダリング時間の推定値であり、前記有効面積が幾何形状データの前記セットのための実面積と疑似面積との和を含み、前記実面積が所定の定数 a_c と等しいか上回る前記幾何形状データ内のポリゴンの表面積に対応し、前記疑似面積が最小ポリゴン・オーバーヘッドの効果に近似し、前記所定の定数 a_c に前記所定の定数 a_c より小さい前記幾何形状データ内のポリゴン数を乗じた値に対応する、レンダリングするグラフィックス・データの特定のセットのための有効面積を計算するステップと、

前記有効面積を前記ピクセル・フィル・レートで割ることによって前記幾何形状のレンダリング時間をリアルタイムで推定するステップと
を含むグラフィックス・システムが幾何形状データの特定のセットをレンダリングする近似レンダリング時間を予測する方法。

【請求項33】 前記グラフィックス・データがピクセル・フィル制限付きであるのに応じてより少ないピクセル計算を必要とする詳細レベルをレンダリングし、前記グラフィックスデータがポリゴン・オーバーヘッド制限付きであるのに応答してより大きいポリゴンを有する詳細レベルをレンダリングするステップをさらに含む請求項32に記載の方法。

【請求項34】 前記実面積を計算する前記ステップが、

a_c が所定の定数であり、 s が g^* の計算のためのスケーリング係数とみなされるモデル空間－画面空間スケーリング係数である、 a_c と s との積より小さいかまたは等しい面積を有するポリゴンによって占められる合計表面積を近似する関数 $g^*(a_c s)$ の値を求めるステップと、

幾何形状データの前記セットのための合計画面面積を事前計算するステップと

モデル空間－画面空間スケーリング係数を計算するステップと、

($1 - g^*(acs)$) 実行数値積分を使用して前記合計面積乗算するステップとを含む請求項 33 に記載の方法。

【請求項 35】 前記推定ステップと前記計算ステップがリアルタイムで行われ、前記判断ステップがオフラインで行われる請求項 32、33、または 34 に記載の方法。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

Inter. Appl. No.
PCT/US 99/03227

A. CLASSIFICATION OF SUBJECT MATTER IPC 6 G06T15/00		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 6 G06T G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 00811 A (RESOLUTION TECHNOLOGIES INC ;DAW CRAIG (US); DEHMLow JAMES (US); E) 8 January 1998 see page 11, line 26 - page 12, line 11; claims 23-37	20,32
A	EP 0 463 700 A (PHILIPS ELECTRONICS UK LTD ;PHILIPS NV (NL)) 2 January 1992 see column 8, line 4 - line 35	1,20,32
A	EP 0 821 319 A (ADOBE SYSTEMS INC) 28 January 1998 see claim 1	1,20,32
A	EP 0 241 091 A (OCE NEDERLAND BV) 14 October 1987 see page 6, line 25 - page 7, line 43	1,20,32
-/-		
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "B" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
28 June 1999		06/07/1999
Name and mailing address of the ISA European Patent Office, P.B. 5518 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tlx. 31 651 epo nl. Fax: (+31-70) 340-3016		Authorized officer Perez Molina, E

Form PCT/ISA/210 (second sheet) (July 1992)

page 1 of 2

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/US 99/03227

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"VIDEO PERFORMANCE ANALYSIS TOOL" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 37, no. 10, 1 October 1994, pages 255-257, XP000475655 * the whole document *	1, 20, 32
A	PATENT ABSTRACTS OF JAPAN vol. 098, no. 005, 30 April 1998 & JP 10 000825 A (FUJI XEROX CO LTD), 6 January 1998 see abstract	1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

page 2 of 2

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int. Application No.

PCT/US 99/03227

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 9800811	A	08-01-1998	AU 3718497 A	21-01-1998
EP 0463700	A	02-01-1992	GB 2245805 A	08-01-1992
			GB 2245806 A	08-01-1992
			DE 69127516 D	09-10-1997
			DE 69127516 T	26-02-1998
			JP 4233086 A	21-08-1992
			DE 69122557 D	14-11-1996
			DE 69122557 T	24-04-1997
			EP 0464907 A	08-01-1992
			JP 4233672 A	21-08-1992
			US 5394516 A	28-02-1995
EP 0821319	A	28-01-1998	CA 2211247 A	24-01-1998
EP 0241091	A	14-10-1987	NL 8600888 A	02-11-1987
			DE 3780494 A	27-08-1992
			JP 2614226 B	28-05-1997
			JP 62294553 A	22-12-1987
			US 4771340 A	13-09-1988

Form PCT/ISA/210 (patent family annex) (July 1992)

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW

(71)出願人 901 SAN ANTONIO ROAD
PALO ALTO, CA 94303, U.
S. A.

Fターム(参考) 5B080 AA13 GA00

5C082 AA01 AA36 BA12 BA46 BB42

CA31 CA81 CB01 DA42 DA87

MM02

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.